KEYFRAME
MAGAZINE
THE PUBLICATION FOR LIGHTWAVE, INSPIRE, AND AURA ENTHUSIASTS
Character Setup with SockMonkey
Perfect Panoramic Photos
Texturing with Vertex Maps
Messiah Muscle Motion
The Next Five Years
OS X: Mac Future
LScript Sliders
Place with computer magazines
0 74470 92596 5
Issue #22 May/June 2001
$9.95 US $14.95 Canada

Why Wait?

Some things are meant for each other. LightWave [6.5] and Apple's incredibly powerful G4 workstations are a perfect example. LightWave's multi-threaded code takes full advantage of Apple's G4 dual-processor workstations, and LightWave [6.5] also provides unprecedented Open GL support for lightning fast performance. As if this wasn't enough, LightWave [6.5] has optimization for the Velocity Engine. LightWave [6.5] is available now for Mac OS 9 and Mac OS X Public Beta. Anyone running LightWave [6] for Macintosh can download the preview release of LightWave for Mac OS X Public Beta today, and will be able to download the final release of LightWave [6.5] for Mac OS X once it is available - free of charge!

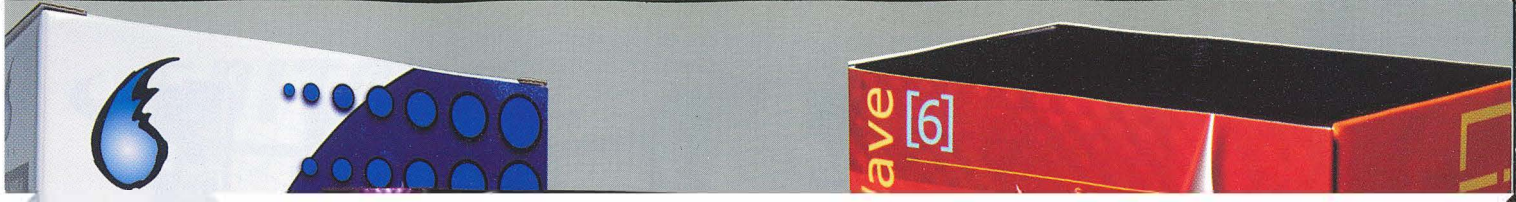
What are you waiting for? Contact NewTek at 1-800-TOASTER to locate the nearest NewTek authorized reseller or visit www.lightwave6.com to learn more about LightWave [6.5] for the Macintosh.

LightWave [6.5]

NewTek

NewTek, 5131 Beckwith, San Antonio, TX 78249
Phone: 1-800-843-8934 Fax: (210) 370-8001 http://www.newtek.com Email: sales@newtek.com
LightWave 3D is a trademark of NewTek, Inc. © Copyright, NewTek 2001 All trademarks and registered trademarks are the sole property of their respective owners.
Screen shot provided by COMPUTERCAFE.

[It's Everywhere]



The Fast,
Powerful Production
Tool for Animation,
Video Paint

and Beyond
Toaster.

MENTION THIS AD AND
GET 5% OFF NEWTEK SOFTWARE!

CHECK OUT OUR GREAT PRICES!

1-800-944-5520 OR 1-702-990-2277

www.dynamic-realities.com

WE'RE TAKING ORDERS FOR
VIDEO TOASTER 2

OF COURSE
OUR PLUGINS
ARE ALWAYS READY TO SHIP!

NATURE FX
REALITOOLS
LUMETOOLS
NAPALM NEW!
IMPACT 2 NEW!

KEYFRAME MAGAZINE

Departments

3 **Editorial** by Doug Kelly
- Continuing Education

4 **Reality Check** by Patty Kelly
- Living with an Animator

6 **On Topic** by David Warner
- VTNT 2.0, Underwater Volumetrics, Win2K, UV Maps

10 **Programmer's Hideaway** by Reiner Schug
- Sliders with LScript

12 **Mac: Up Close and Personal** by Cailean Babcock
- OS X and the Future of Animation on the Mac

32 **Final Cut** by Steve Pugh
- The Next Five Years

Tutorials

15 **Texturing with Vertex Maps** by Barrett Meeker
- Tips for New L6.5 Features

18 **Advanced Character Setup Using Sock Monkey** by James Jacobs
- Character Rigging

22 **Messiah Muscle, Part I** by Ron Wolfe
- Using Muscle Motion in Your Animations

26 **Panoramic Photography Made Easy** by Jared Bendis
- Creating Perfect Environment Maps

Cover

The cover image is an all-CG shot from the "Memo's Crash" sequence in the feature film DRIVEN, with visual effects by Pixel Magic. The scene was rendered in LightWave 6.5 and composited with Adobe After Effects. Every element in this shot is computer generated, including the racetrack itself (which was based on photographs of the real location), the cars, super-slow-motion rain, sparks, and smoke. The end result consisted of more than 100 compositing layers. Picture supplied by Pixel Magic (**http://www.pixelmagicfx.com**)

Editor
Doug Kelly *dougk@keyframemag.com*

Advertising Rep
Charles Edgin *charles@keyframemag.com*

Co-Conspirators, Writers, and Artists
Cailean Babcock *cailean@provalue.net*
Jared Bendis *jaredjared@earthlink.net*
James Jacobs *jamesj@compt.com*
Patty Kelly *dakelly@megalink.net*
Barrett Meeker *gluge@gluge.com*
Steve Pugh *steve@foundation-i.com*
Reiner Schug *diver@interialabs.de*
David Warner *dave@tv3d.com*
Ron Wolfe *rwolfejr@woh.rr.com*

Proofreader
Patty Kelly

The Digital Underground
DMG Publishing
2756 N. Green Valley Pkwy. #261
Henderson, NV 89014
702-990-8656

Subscriptions & Back Issues
Order online at www.keyframemag.com
Email sales@keyframemag.com
1-888-778-WAVE

Keyframe Magazine is published 6 times annually by DMG Publishing.

© 2001 DMG Publishing. All rights reserved. Reproduction, in part or in whole, without written permission is strictly prohibited. Printed in the USA.

The Secret Formula
Keyframe Magazine is created using Quark 4.1, Photoshop 5.0, LightWave 6.5, and various plug-ins.

No dongles were harmed in the making of this publication.

EDITORIAL

CONTINUING EDUCATION

By the time you read this, I will be immersed once again in the demanding and rewarding environment of higher education. As I've done for the past few years, I'll be teaching seminars or summer classes at various schools. Last year's seminar in Malaysia was fun but exhausting, so I'll be sticking a little closer to home. Just the U. S. East Coast this time, thanks very much, all destinations being driveable in case the airlines make good their veiled threats regarding the effect of fuel prices on airfares.

I enjoy teaching. It sounds trite, but it's true that good teachers learn from their students just as much as the students learn from their teachers. As a teacher, I always come away from a class with more information than I had going in. However, that new information may not be on the topic I was teaching. I keep an especially sharp eye out for data on students' base capabilities, attitudes, expectations, and in-class and out-of-class behavior. The students (I hope) don't realize they are guinea pigs for my observations and experiments, but then again I may be a test subject for their own mental laboratories. It's a fair swap.

One of the reasons I enjoy teaching is that it forces me to reexamine my own theories, beliefs and habits. Students have a refreshing and invaluable habit of questioning nearly everything. Even if that questioning doesn't extend beyond, "Do we *have* to do this?," at least it forces the teacher to justify the assignment. "No, you don't have to do this. But if you don't learn it now, you'll look like an idiot the first time your supervisor (in a real job) asks you to do the same thing."

One of the downsides to teaching is experiencing the sinking sense of impending doom when dealing with students who just don't have what it takes. Persistence, intelligence, talent, common sense; a lack of any one of these attributes will inevitably be fatal to an artist's career in this industry (but you can always become a producer). Persistence can be learned through experience, as can common sense. Learned and practiced skills can make up for a great deal of talent. Ignorance isn't a mortal flaw; it can be corrected with education. Unwillingness to learn, however, is always and inevitably fatal. Learn or die.

Whether you are a working pro, a student, or a hobbyist wannabe, continuing education is your lifeline and your path to the career you want. If your career is your dream, then education is a crucial part of that "pursuit of happiness" named in our Declaration of Independence.

A common rule of thumb is that working professionals should spend ten percent of their time on continuing education. If you're punching a timeclock, that means you should average four hours a week reading Keyframe or similar journals, perusing FAQs, sifting through listserv traffic, reading books, viewing multimedia CDs, or taking classes. If you're as busy as most people, you want the most bang for your buck, in both time and money.

With the number of tutorials and other information packed into this issue, I believe you'll find many hours of continuing education between these covers. I hope you'll consider Keyframe a valuable part of your own continuing education.

Doug Kelly

Living with an Animator

by Patty Kelly

Sometimes the scent of freshly baked homemade bread mingles with the odor of firing Sculpey in my kitchen. Even though my profession focuses on working with blind people, the bookcases in my home are heaped with material on visual images. I know what an alpha channel is. My spouse's idea of the perfect date night includes going to G-rated children's movies and then driving through fast food restaurants to obtain the toys associated therewith. Yes — I am married to an animator.

I have always been a list-maker, so here are three important lists for those contemplating living with an animator: Five Downsides to Living with an Animator, Four Upsides to Living with an Animator, and Three Truths About Animators that the Rest of Us Must Accept. Let's begin with

Five Downsides to Living with an Animator:

1. The animator goes on-line at the strangest hours and for the most frustrating periods of time; the animator will claim it's for "research," but to me it seems a lot more entertaining than the research I did in graduate school. No professor at my university recommended DejaNews as supplementary reading material.

2. New laserdisks will start popping up in your home with titles like "Computer Animation Festival" and "Heavy Metal" — whatever happened to a good, old-fashioned romantic comedy?

3. Animators who work from home can do so on their own schedules and in whichever manners they find most comfortable. This may include wearing the most bizarre outfits imaginable (although a black tee-shirt with a software company's logo is de rigeur as part of the ensemble), rolling out of bed in the wee hours of the morning to animate when inspiration strikes, working on a laptop at a family reunion (Fortunately, in my case, it involved his side of the family, so aside from much eye-rolling, I didn't have to deal with much in the way of negative reactions.), periods of round-the-clock work followed by periods of virtually catatonic sleep, and eating patterns designed to stand the FDA food pyramid on its pointy little head.

4. Strange-looking and expensive-seeming equipment will appear in odd places throughout your home. The camera and tripod near the bedroom door made me think our marital relations were going to get a little spicy 'til I realized the rig was pointed the other direction, toward a maquette taped down to a plant stand under the window — at that point, I was more concerned with where my ficus had ended up than with what negligee I'd wear at bedtime.

5. Your refrigerator will be cluttered with containers of caffeinated beverages. Others living with animators complain about the perpetual pot of coffee. Or the bottomless dish of jelly beans. It all boils down to the same thing: legal, moral chemical stimulants. Animators gotta have 'em, and you'll quickly learn your own animator's stimulant-of-choice. Personally, I am seriously considering buying stock in the company that manufactures Dr. Pepper.

Let me not lead you into thinking that living with an animator is all bad. Here is a list of

Four Upsides to Living with an Animator:

A. I never have to worry about having families with young children drop by to visit unexpectedly — there are plenty of toys around and we have lots of animated children's videos. It is worth noting here that no actual children reside in our home.

B. Because my husband works by himself and glued to a computer all day, I never worry that he's having an affair. I do not feel threatened by Lara Croft. The only opportunity he has for infidelity is his annual sojourn to SIGGRAPH; since my general (and admittedly both biased and based-on-no-personal-experience) impression of this event leads me to believe that the activities there include such things as literally drooling over techno-gadgets and competing for the title of King/Queen Geek, I am not worried about those trips.

C. Our dogs are very happy. In the wild, dogs live in packs in which they basically lie around sleeping most of the day and get excited only at mealtimes or when there's an intruder on the pack's territory. At our house, the dogs can lie around in peace and quiet all day (They've learned to ignore the soft clicking of the mouse and keyboard.). Mealtimes are the highlights of their day, as are those occasions when the Federal Express guy or UPS driver invade their territory, at which point dogs and animator alike race about in a frenzy. As far as the dogs are concerned, animators follow canine behavior more closely than most bipeds they know.

D. Living with an animator is akin to having a live-in computer consultant. Since animators work with computers all day, they usually know what those alarming pop-up windows and cryptic error messages actually mean and (most importantly) how to make them go away without irreparably harming your system. Not only that, but thanks to your animator, you will have the coolest desktop wallpaper

and most amusing alert noises in your office building!

Now there are certain characteristics of animators that are intrinsically neither good nor bad; they simply are. Here are

Three Truths About Animators that the Rest of Us Must Accept:

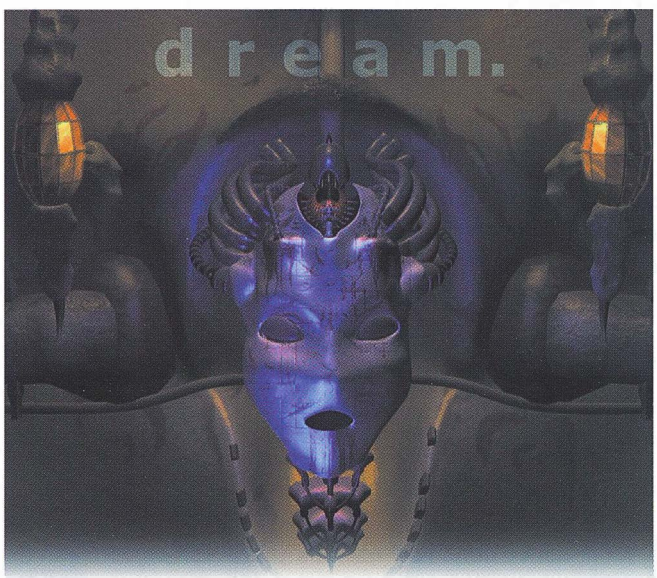
I. *An animator has a job that can be hard to define or explain to people.* I dread that inevitable moment at a cocktail party when the perky Junior League wife of the community's reigning legal-eagle hotshot crown prince lawyer asks sweetly, "...and what does your husband do?" Simply replying, "He's an animator," isn't enough. I find myself rambling on and on along the lines of "...you know, like in *Toy Story* or *A Bug's Life*..." and being met with a glazed-over look or (even worse) an artificial smile superimposed on a "you-poor-little-thing" face. I once thought I could bypass all that explanation about character animation and so forth by answering, " He edits 'Keyframe Magazine'." The slightly deaf elderly lady who asked the question at a church potluck turned to her even-harder-of-hearing companion and shouted, "He writes a magazine about keychains — you know, for all those people who collect keychains!" Having neither the desire to disrespect an elder nor the technical expertise to explain what a keyframe is or does, I simply snagged another devilled egg off the buffet table, smiled, and ducked away. Since then, I've learned that if I mumble something about "computers" in a vague way, the response is much more positive. But I always wonder and worry whether I'm denigrating the profession of animation by not being more specific.

II. Don't be fooled into thinking, because animators spend so many hours in front of computer screens, that their primary strengths are mathematical or verbal. Oh, yes, they do have skills in those disciplines. However, *an animator's driving force is an artistic temperament.* They honestly can't control the creativity that oozes from their every pore. One would never expect a professional statistician, for example, to have wildly fluctuating moods concerning the shape her work is taking; however, this is pretty much standard operating procedure for an artist. Although dealing with the vicissitudes of the artistic personality can be something of a trial, look at it this way: without having to apply for a grant or to come up with an endowment, you've got a permanent artist-in-residence...right in your very own home!

III. *Animators belong in Never-Never Land.* No, I don't mean they're crazy; I just acknowledge that they are adults who never grew up. They like slapstick and visual gags. They like to take things apart to see how they work. They spend so much time interacting with machines that their social skills in dealing with other human beings may leave a bit to be desired. They can watch the same video over and over and over and over. In these ways, that animator standing before you is more like a child than like an adult. Once you have accepted the child-like attributes of animators, things can go more smoothly. For instance, I no longer spend time and money seeking expensive gifts to bring home when I go on business trips and then feel hurt when they're not used. Instead, I think about that child-like quality when choosing gifts. Recently, for example, I returned from a trip to Pittsburgh with a motion-sensor growling Komodo dragon statue which has been enthusiastically received by my animator and which has been appointed a choice spot in the studio/office. Don't put up with childish behaviors (temper tantrums and so forth), but do celebrate the child-like delights enjoyed by animators.

Once I accepted these three truths about my husband as an animator, it was easier to remember that I love him for the person he is. Being an animator is part of who he is — it's not just what he does for a living. (I'm convinced that most animators would continue animating even if they were guaranteed never again to make even a thin dime for their efforts.) Since I love the rest of him, I've gotta take the animator part, too. Frankly, I'm glad to have all of him. May you find as much happiness and joy in living with your animator as I have in living with mine.

Patricia K. Kelly, Ed.D., is a consultant in adult education and blind rehabilitation residing in Maine. She can be reached through the animator with whom she lives at ***dakelly@megalink.net****.*



3D Exchange
Animation Training
Premier Authorized Training Center
NewTek
www.3dexchange.com • (510) 747-1540

VTNT 2.0, Underwater Volumetrics, Win2K, UV Maps

by David Warner

NewTek made a very impressive showing at this year's NAB in Las Vegas. While they did not exhibit anything new for LightWave, they did unveil a new version of their Video Toaster NT system. This is a product that will add a tremendous amount of power and efficiency to the LightWave work environment.

Video Toaster NT 2.0 merges the power of LightWave and Aura with an amazing video editor/switcher system that has all the sparkle and excitement of the original Video Toaster. Aura and LightWave already complement each other nicely, but VTNT 2.0 looks like it will really tie everything together, and it will give users an impressive amount of power.

I'm sure there will be a lot of press about the new Toaster's capabilities. Some of the things that impressed me most about the few VTNT 2.0 demos I saw were the pixel-tracking capabilities of Aura and how those data could easily be fed into LightWave. When these two programs are integrated with the non-linear editing capabilities of the Toaster, you end up with a very slick and efficient working environment. Add to that the bluescreen and rotoscoping capabilities of the Toaster, plus all the other cool things it can do, and you're suddenly talking about a pretty cool piece of hardware. I know I am certainly looking forward to its release late this year!

I'm going to squeeze in as many questions and answers as possible, so let's get started.

Q I need to simulate beams of light shining through water but I'm not sure where to get started; can you offer some suggestions?

A This can be done very easily with a Volumetric light that has a projection map applied to it.

The first thing you need to do is to create a projection map image sequence for your volumetric light. You can do this by applying the Underwater procedural texture as a color map to the surface of a flat plane. Use high contrast colors, such as black and white, to make the procedural texture stand out nicely on the flat plane object. The Underwater procedural is a repetitive texture, so if you input a Wave Length that is cleanly divisible by the Wave Speed, you will end up with a precise number of frames to render for a looping image sequence. In other words, if you enter a Wave Length of 0.1 and a Wave Speed of 0.025 then the image sequence will repeat every 40 frames. You should definitely render out an image sequence that loops; otherwise, you will end up with a stutter in your underwater sparkle effect.

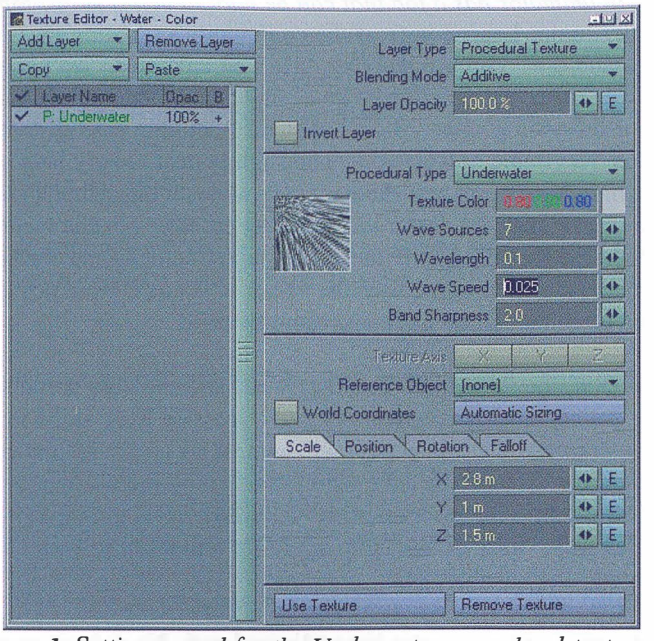


Figure 1: *Settings panel for the Underwater procedural texture.*

Once the image sequence is rendered out and stored on your hard drive, you can apply this sequence as a Volumetric Projection map to a light in your scene. You should position this light very high above the scene, shining down on all the objects in your scene. You will have to change the light type to Spotlight if you want to use the Projection Map options. You will also have to enable the Volumetric Lighting option so that the

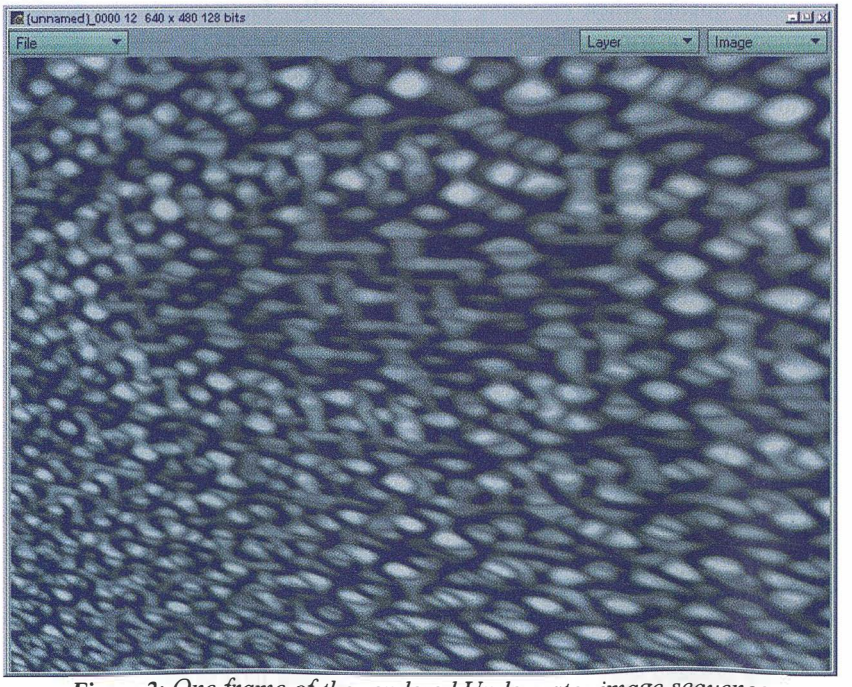


Figure 2: *One frame of the rendered Underwater image sequence.*

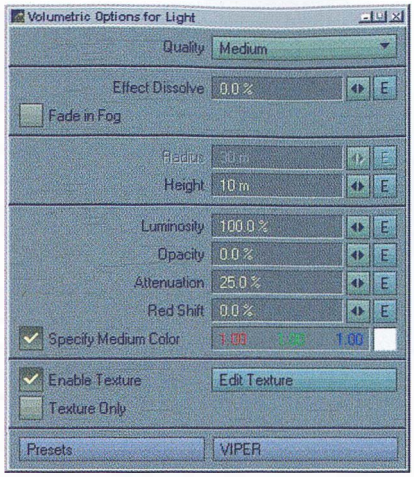


Figure 3: *The Volumetric Lights settings panel.*

Projection Map image sequence will generate the beams of sparkling light effect.

One advantage of using a Spotlight this way is that you can use Shadow-Mapped shadows to reduce render times and to increase realism in your scene. However, there are a few things you must remember to do before you render. Most important of all is that you must turn off all shadow casting options for your water surface object. You must do this because shadow-mapped lights do not recognize transparent objects, so your water would cast a very big, black shadow on everything in the water.

Another important final detail is to adjust the cone angle of your shadow-mapped spotlight so that the underwater texture image sequence looks good on the water surface. By reducing the cone angle size, you will also reduce render times and make the shadow-mapped light look better.

Last but not least, you will probably want to add a fog effect to the scene to simulate the density of an underwater scene. The end result of all this should be nice beams of volumetric light streaming down through the surface of your water object. With the animated underwater procedural texture applied as an image sequence, the water surface and the volumetric light beams will move, and any objects passing through these beams of light will appear to have sparkles of light on them. An added bonus of this setup is that the shadow-mapped light will cast shadows on your objects and the volumetric light beams will also be shadowed as objects pass through them.

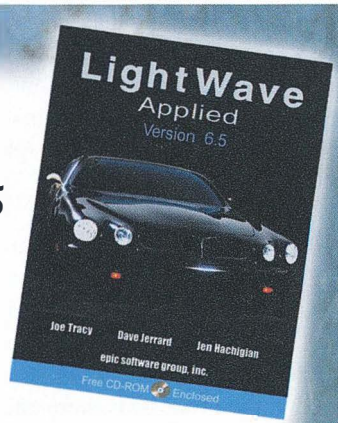
Q Is there a way to merge previously saved projects in LightWave? I am trying to import camera and light setups from scene to scene.

A If you use the Load From Scene option, you can import the lights from another scene. You can also save the camera's motion path from within LightWave's graph editor. However, neither of these options are particularly efficient. A much better way to work in LightWave is to set up a default scene that has the camera and light settings that you use most

Pre-Publication Offer

Coming July 2001

LightWave™ Applied
version 6.5



From the industry-leading experts that brought you LightWave 3D™ Applied, Version 5.6...
Imagine a tutorial that has over 100 steps and more than 100 images, and that's just one chapter! ***LightWave™ Applied—Version 6.5*** teaches readers from scratch how to master LightWave™ 6.5 through the development of actual projects. Readers will be treated to in-depth tutorial chapters on using LightWave™ 6.5 features to build items like buildings, cars, mountains and more. Readers will learn top cel shading techniques to make a 3D character look hand drawn when rendered. Also covered is how to prep a character for animation. From IK and Bones techniques to creating an entire architecture landscape with camera fly-throughs, ***LightWave™ Applied—Version 6.5*** gives in-depth coverage to techniques critical for animation success.

Excellent guide for instructors teaching computer animation.

Features of LightWave™ Applied—Version 6.5 Include:

- Twelve step-by-step, hands-on tutorial chapters to help you construct detailed and realistic objects, many which you'll also learn to animate
- A specially designed Website that allows you access when you purchase the book to help answer any questions
- A FREE CD-ROM packed full of material from the book along with bonus items such as an updated version of "Breaking into Hollywood"
- A special introduction section for beginners to LightWave™ that will familiarize you with basic functions of Modeler and Layout
- A special chapter teaching readers how to start their own local animation company
- Insight from industry leading authors along with their tips and secrets to help you succeed

$44.95 if purchased before June 15th

$49.95 Regular Price,
Over 600 pages, Soft Cover, Item #POSB021,
FREE CD-ROM

Call 1-800-598-6008
Outside the U.S. 1-218-723-9180
Fax 218-723-9146
Visit and order on-line at
www.advanstarbooks.com/lightwave/
Please mention code O-KF when ordering

961442

often. You can easily create multiple scene files for commonly used setups such as outdoor daytime lighting, indoor lighting, underwater, etc. Then you just load up one of these default scene files whenever you are about to start a new project, and much of lighting and camera setup is already taken care of.

Q I recently upgraded my PC to use Windows2000 but as soon as I did this, LightWave could not find the hardware dongle anymore. I tried everything I could think of but I still can't get LightWave to work with Windows2000. What should I do to fix this?

A The problem you're experiencing is that the system drivers for the hardware dongle are not fully compatible with Windows2000. You need to update a few things in your system in order for LightWave to work correctly again.

Since Windows2000 is still a fairly new operating system, the most important thing you need to do is make sure that you have the newest upgrades and service packs installed. This is true for all operating systems, because chances are very good that the version you installed from a CD-ROM is not the same as the newest version available via the Internet. In this case you should go to the MicroSoft.com website and download any service packs that are available for Windows2000.

Once you have your operating system properly upgraded, you need to download a Windows2000 compatible version of the Sentinel system drivers from the Rainbow.com website. Rainbow is the company that makes the hardware dongle for LightWave, and they provide downloadable upgrades of their Sentinel System Drivers. Once you download and install these updated system drivers, your LightWave hardware dongle will be visible to Windows2000 and LightWave should boot up without any problems.

Q I'm trying to clone a large number of objects that have UV maps applied to them. There will be hundreds of these objects and cloning them manually is just not an option. I found a plug-in that does a great job of cloning and offsetting the position of all these objects, but the UV map information gets corrupted in the process. Is there another way to do this?

A Yes, there is a great plug-in called "Clonez" that does exactly what you want. It does not corrupt the UV map data, and best of all, it's free! The best way to find this plug-in is to go to www.flay.com and do a search for the plug-in's name. Once you download and add this plug-in to LightWave, you'll be cloning objects to your heart's content.

Q I'm trying to follow some tutorials that were written for LightWave 6.0 but the interface layout is very different in the newest version of LightWave 6.5. Is there any way to switch the appearance of the interface back to the L6.0 layout?

A The layout of menu buttons changed significantly in LightWave 6.5. However, one of the really cool things about

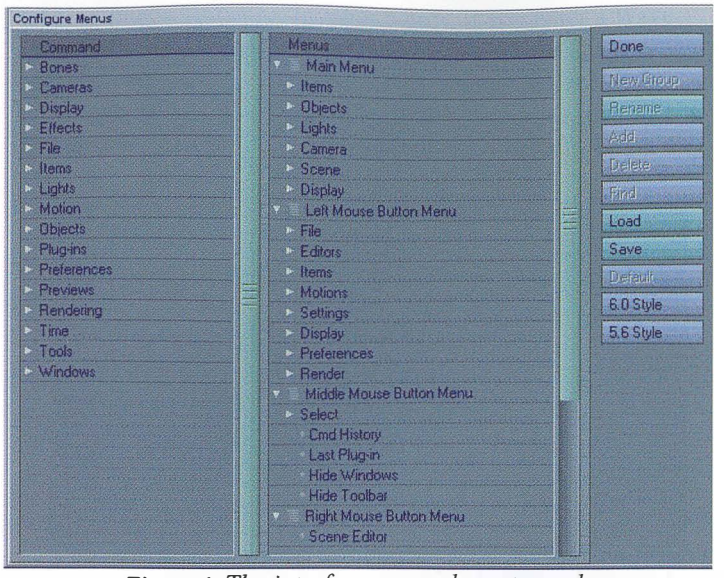


***Figure 4:** The interface menus layout panel.*

the newer versions of LightWave is that you can totally customize the menu layouts. There are a few presets available for easy access, including a version 6.0 layout, which is more compatible with many of the tutorials that are available commercially and on the net. To access these presets, or to create your own, go to the Layout pull-down menu, highlight "Interface," and then select the "Edit Menu Layout" option. This will bring up a panel that gives you easy access to all of the functions and plug-ins that are part of LightWave, as well as give you the ability to create custom buttons and menus for any commercial plug-ins that you have purchased. You will also see that there is a "6.0 Style" preset, which will enable you to quickly reset the menu layouts back to the way buttons were arranged in LightWave version 6.0.

Q How do I add or remove polygons from a UV map?

A To remove polygons from a UV map, start by selecting the relevant UV map, select the polygons you want to remove, then click on the "Clear Map" function from the Map menu. If the polygons are positioned at the edge of a section of the UV map, you may first have to Un-Merge those polygons before removing them from the UV map. Be sure to re-Merge the polygons in your object when you are done. To add polygons to a UV Map, the process is pretty much the same as described above, but you will use the "Make UVs" option instead.

Q I created one half of an object, applied a UV map to it, and then mirrored the results to create the other half of the object. I merged points and everything but the UV map was not mirrored correctly. What happened and how do I correct this?

A In Modeler version 6.5 there is a new "Transform UV" option in the Map menu that will enable you to adjust your UV maps in a number of ways. In this case, you would select the mirrored polygons and then select this "Transform UV" function. In the panel that pops up, enable the Scale values and change the U value to -100%. This will flip the UV map that

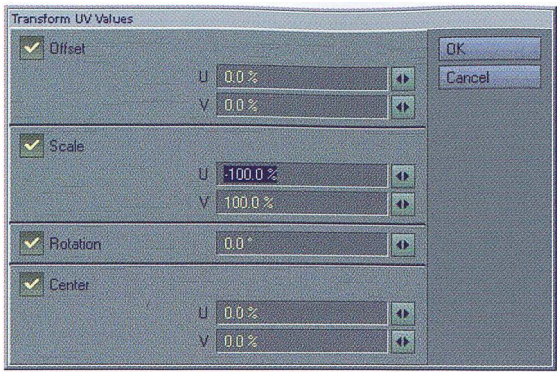


Figure 5: *The Transform UV settings panel.*

is affecting that group of polygons.

I am running LightWave 6.5 on a Mac and I need to import some TrueType fonts for a logo object that I'm trying to create, but I don't see any way to do this. Is there something wrong with my copy of LightWave or am I missing something?

Unfortunately, the Mac version of LightWave 6.5 does not yet recognize TrueType fonts. It will work fine with Postscript fonts, so if you can find the same font in a Postscript format you will be all set. There is a new version of LightWave 6.5 that should be available by the time you read this, and I hope that upgrade will fix this problem.

One way around this problem is to use a program like Adobe Illustrator, CorelDraw, Freehand, or any other program that allows you to type in text and then save it in EPS format. The text should be converted to Outline or path mode before saving; but if you do this correctly, then Modeler will load the file without any problems and you will have your TrueType font logo imported. It's an awkward work around, but until NewTek is able to correct this problem in a future upgrade, it's a better option than not using the font at all.

I am trying to render a small droplet of water that is lying on another object, but the shadow being cast by the droplet is far too dark. Is there any way to brighten up shadows on transparent objects like this?

It sounds like you are using a Shadow-Mapped light, which creates great looking soft-edged shadows, but does not recognize transparent objects, so even a totally clear pane of glass would cast a completely dark shadow. There are many ways to solve this problem so I'll just cover the best ones and you can figure out which one works for your situation.

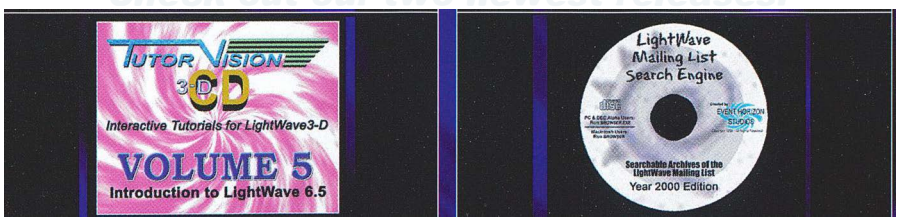
Continued on Page 30

TUTOR VISION 3-D CD

The BEST WAY to learn LightWave!

Easier to understand than a book...more tutorials than any video-tape!

Check out our two newest releases!


TUTOR VISION 3-D CD
Interactive Tutorials for LightWave3-D
VOLUME 5
Introduction to LightWave 6.5
LightWave Mailing List Search Engine
Searchable Archives of the LightWave Mailing List
Year 2000 Edition

TutorVision3-D Volume 5 is our first set of tutorials for the new LightWave v6.5!

Available NOW for $49.95

LightWave Mailing List Search Engine CD Year 2000 Edition

Available NOW for ONLY $9.95!

FREE Sampler CD!

This FREE SAMPLER CD introduces you to the TutorVision3-D series of CD-based video tutorials. This CD contains 1 complete video and text tutorial from each of the four currently available volumes of TutorVision! It also contains 2 brand new video tutorials not found on any other TutorVision3-D CD!

That's 10 video and text tutorials for FREE!

So how do you get your FREE Sampler CD?

Just get your address to us via e-mail, phone, or use the special form on our web-site, and we'll mail the Sampler CD to you right away, with no obligation to buy anything!

Quantities Limited - ORDER NOW!

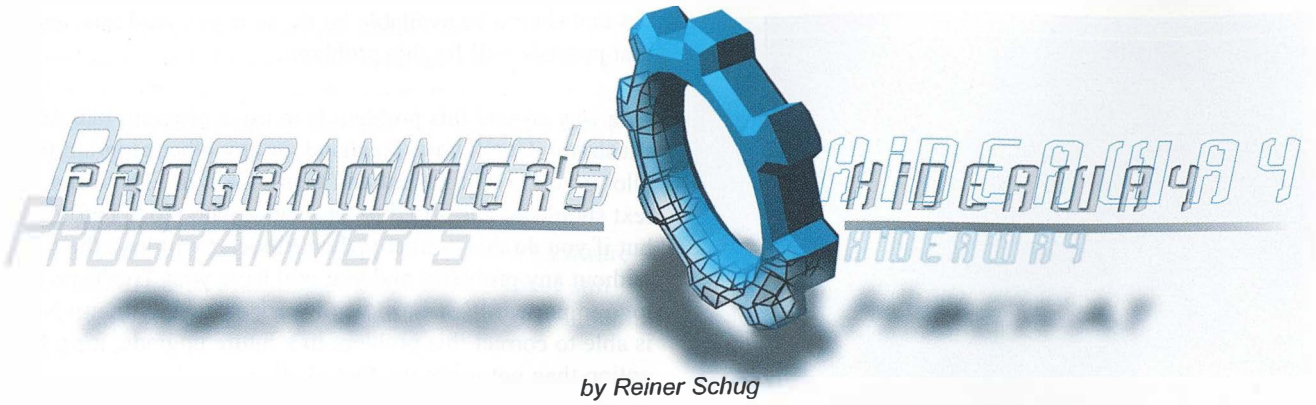
NEW LOW PRICES on all of our "older" titles!!

We've made BIG price reductions on our first 5 CDs! Check out our web-site or give us a call for complete details and pricing!

Event Horizon Studios

Phone: 518-756-6850
FAX: 518-756-6836
E-Mail: info@tv3d.com
Web-Site: http://www.tv3d.com

Sliders with LScript



by Reiner Schug

Sliders can be used as powerful tools that enable you to enhance your animations and speed up your workflow. A slider panel is like a programmable infrared remote control that you can use to operate a TV, a CD player and a VCR, all from a single handy device. With a slider panel, you can modify many animatable attributes from one compact panel. Of course, you could also create a channel set in the graph editor and control the values from there. However, if you have different channel types or value ranges, you have to do a lot of zooming and moving around in the motion graph. A properly coded slider panel can greatly simplify the complex tasks of animating a number of widely variable attributes.

Let's have a look at the LScript slider control:

`ctlslider(title,default,minimum,maximum)`

A slider has a fixed width of 100 pixels, so the maximum control resolution is 100 ticks. The return value of the slider is an integer. If you want to manipulate a floating point attribute with a slider, you have to calculate the value by multiplying the Min/Max range and then dividing the return value of the slider accordingly.

Code Selection 1 shows the code for a simple slider panel with 3 sliders. Ambient1 and Ambient2 both control the Ambient Light Intensity in the range between 0% and 50%.

Load this script into Layout as master script, and open the Global Illumination Light Panel to see the difference between the sliders. Ambient1 allows you to change the value in 0.5% steps, and Ambient2 changes the value in 1% steps. The third slider is simply a frame slider. Since frames are integer values, no calculation is needed in the third slider's callback function. Just pass the value to "GoToFrame".

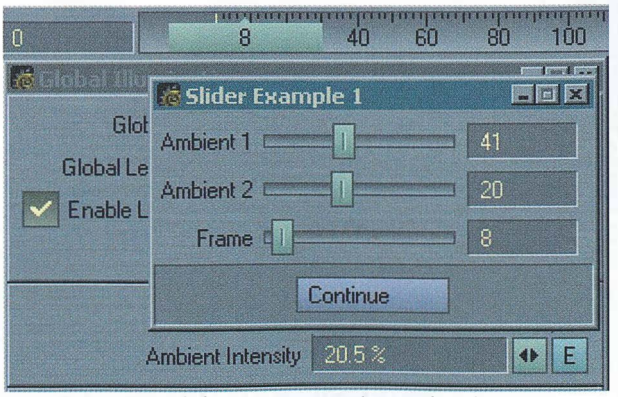


Figure 1: *3-slider panel created by code selection 1.*

```
// Code1
@script master
c1..3;
options
{
        if(reqisopen()) { reqend(); return; }
        reqbegin("Slider Example 1");

        c1=ctlslider("Ambient 1", 0, 0, 100);
        c2=ctlslider("Ambient 2", 0, 0, 50);
        c3=ctlslider("Frame", 0, 0, Scene().previewend);

        ctlrefresh(c1,"ref_c1");
        ctlrefresh(c2,"ref_c2");
        ctlrefresh(c3,"ref_c3");

        reqopen();
}

ref_c1: value { AmbientIntensity(value*50/10000); }
ref_c2: value { AmbientIntensity(value/100); }
ref_c3: value { GoToFrame(value); }
```

Code Selection 1: *LScript to create a simple 3-slider panel.*

Keep in mind that the slider is limited to 100 ticks. To take advantage of the maximum slider resolution, the range between minimum and maximum should always be higher than or equal to 100. If you want to use a slider for a rotation channel, you will have to convert the slider value from degrees into radians using the formula:

`newvalue=rad(value)`

Now let's take a look at channels. A channel's key values are always floating point numbers, and a channel can be one of four types:
NUMBER: e.g. Scale.X,
DISTANCE: e.g. Position.X
PERCENT: normally a value between 0 and 1 (0%-100%), e.g. Light-Intensity
ANGLE: angle in radians, e.g. Rotation.X

The **ctlchannel** control returns a channel object which is used to manipulate keys with the following methods:
Create a new key: key = channel.createKey(<time>,<value>)
Delete an existing key: channel.deleteKey(<key>)
Set the incoming curve type: channel.setKeyCurve(<key>,<shape>)
Check if a key exists at the current time: key = channel.keyExists(<time>)
(You can find more information about the channel object and its

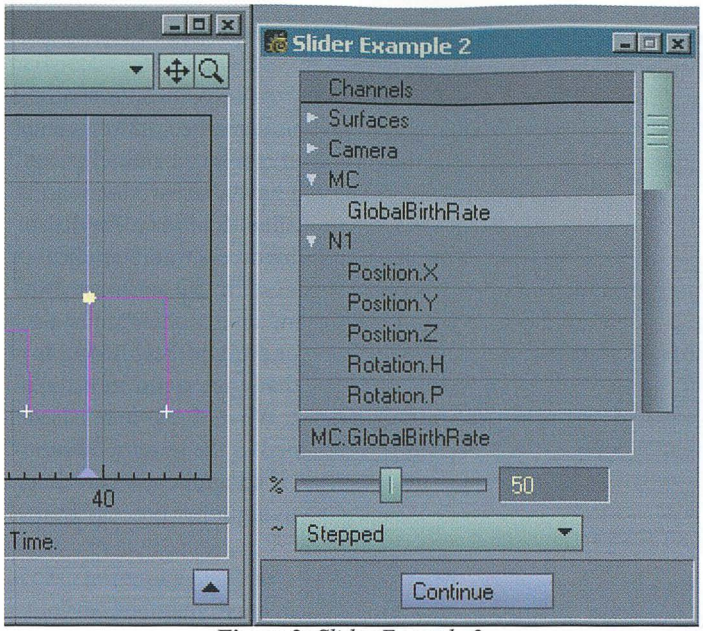


Figure 2: *Slider Example 2.*

methods in the LScript V2.1 release notes.)

Now let's have some fun. Create a master channel of type Percentage, then open the graph editor and select it. Load the script in Code Selection 2 and select the master channel in the channel control. As you drag

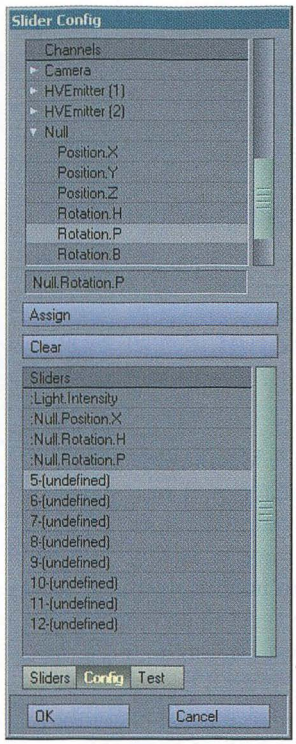


Figure 3: *Slider Interface.*

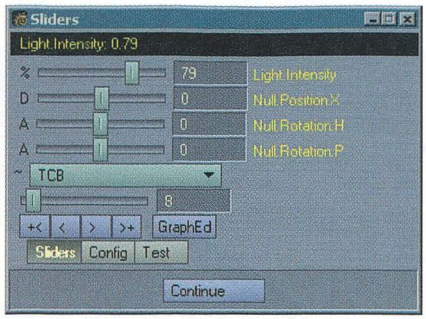


Figure 4: *Slider Interface.*

the slider or change the key shape you can see the curve changing in the graph editor. When you move the frame slider to another frame and then drag the slider, a new key is created. You could now link the master channel with ChannelFollower to the "BirthRate" channels of all the particle emitters in a scene, Figure 2.

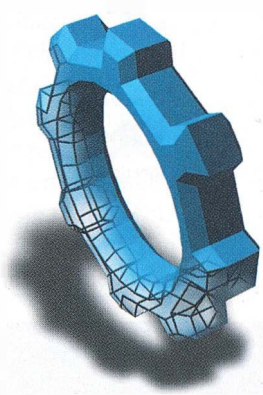
So far, all the sliders we have done were "hard-coded". To create a nifty slider panel we must be able to add and configure the sliders as we need them during the animation process. Figures 3 and 4 are some screenshots of this Interface. If you want to know how this is done you can download the complete script from my website (see URL below).

You can find a collection of my freeware LScripts here at
http://interialabs.de/lw/lscript

Reiner Schug is 29 years old and lives in Bavaria, Germany. His first computer was a C64, then an Amiga; now he is mostly using Windows NT. Reiner has studied programming and first got in touch with LightWave with version 5.0. He now works as a Flash, C/C++, and HTML programmer for various web and multimedia companies, especially ***www.viewsion.de****, and also worked on some 3D and multimedia productions using LightWave 3D. If you need a special script or want to hire him as a coder or animator, you can email Reiner at diver@interialabs.de.*

```
// Code2
@script master
c1..3;
keyShape=1;
Shapes=@"TCB", "Hermite", "Bezier", "Linear", "Stepped"@;
ShapeID=@CHAN_TCB, CHAN_HERMITE, CHAN_BEZIER, CHAN_LINEAR, CHAN_STEPPED@;
selCH;
options
{
        if(reqisopen()) { reqend(); return; }
        reqbegin("Slider Example 2");

        c1=ctlchannel("Channels",250,10,selCH);
        c2=ctlslider("%", 0, 0, 100);
        c3=ctlpopup("~",keyShape,Shapes);

        ctlrefresh(c1,"ref_c1");
        ctlrefresh(c2,"ref_c2");
        ctlrefresh(c3,"ref_c3");

        reqopen();
}

ref_c1: value { selCH=value; }

ref_c2: value
{
        thisKey=selCH.createKey(Scene().currenttime,value/100);
        selCH.setKeyCurve(thisKey,ShapeID[keyShape]);
}

ref_c3: value
{
        keyShape=value;
        if( (thisKey=selCH.keyExists(Scene().currenttime)) != nil )
                selCH.setKeyCurve(thisKey, ShapeID[keyShape]);
}
```

Code Selection 2: *Creating the button functions.*



Up Close and Personal

by Cailean Babcock

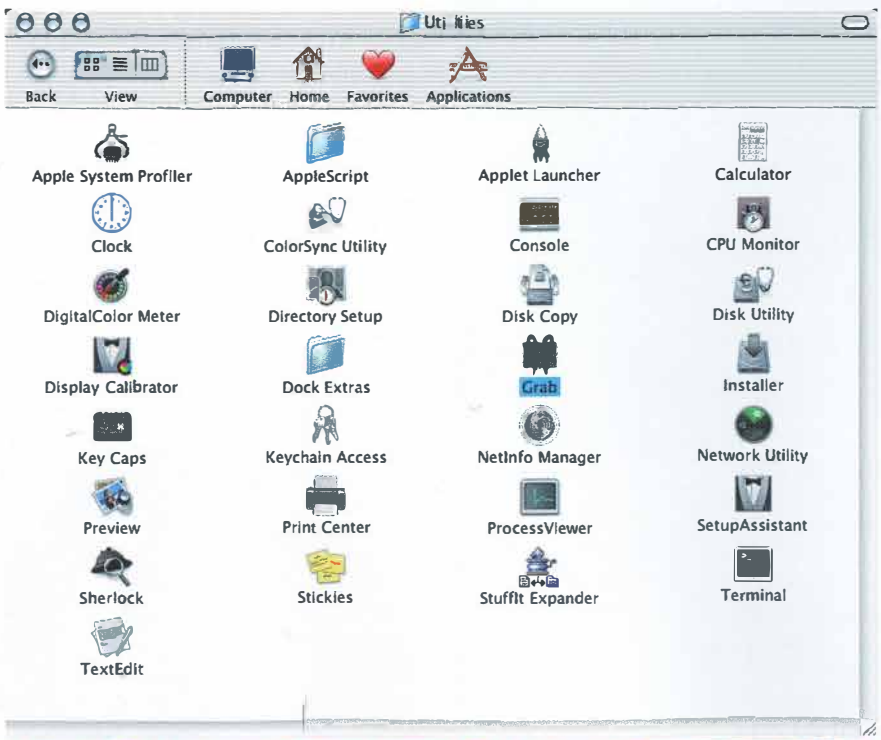
Since its introduction in 1984, Apple's Macintosh computer has enjoyed one of the most colorful reputations of any computer system ever invented. Often loved, sometimes despised, but accepted as the innovative force in desktop computing, the Macintosh has managed to hold its own in the face of the most challenging odds. Apple is set to do it again with the unveiling of the latest and greatest offering to the Macintosh world: OS X. Three years and several false starts in the making, Apple's answer to high-end workstation capability and the sophistication of modern computing needs has finally become a reality.

With the advent of OS X, Apple has elevated the Mac from desktop computer to powerhouse workstation, and in ways that Mac users have only dared to dream of. As a matter of fact, the very phrase "desktop computer" has become rather inappropriate for the Macintosh under OS X. This isn't just your standard Mac evangelizing, either. The underpinnings of the Mac OS subsystems and core architecture haven't just been revised; they've been replaced. The only thing that remains consistent is the interface; even that has experienced some serious changes, but all for the better.

You've heard phrases such as symmetrical multiprocessing, preemptive multitasking, and protected memory bandied about for months (but I'm willing to bet you haven't heard the word "bandied" much). What does this translate to in plain English? Improved performance from existing hardware, more applications running interference-free, and a system that's almost crash-free. Gone are the days of quitting an application just so that you can assign it more memory for a big project. RAM is now dynamically allocated, meaning that your applications use the RAM they need when they need it. When RAM is released by an application, it is promptly returned to the system. The RAM itself is protected, meaning that if an application crashes, it does so in its own secured space, leaving other processes untouched.

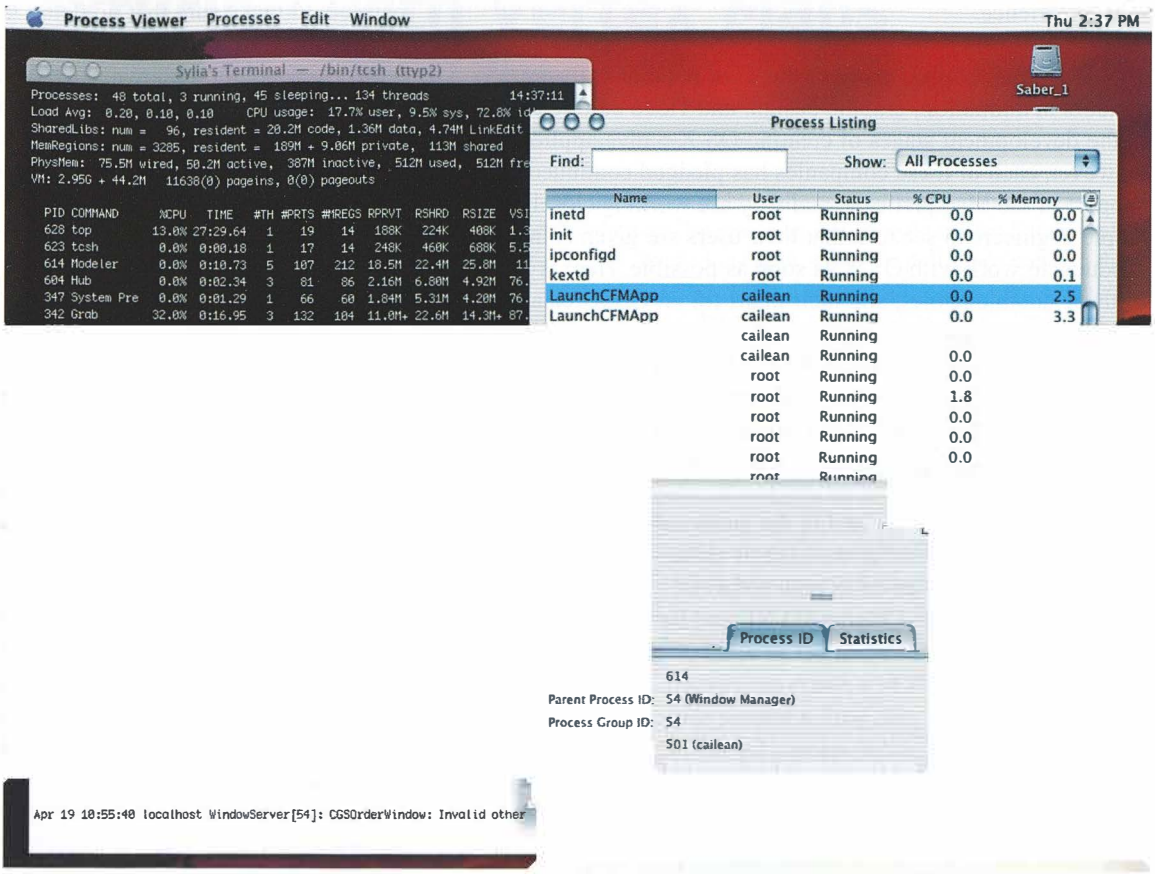
Okay, we have a system that's fast, crash resistant, and capable of leaping tall tasks in a single nanosecond. Does it still work for the artist? You bet, and it does it with style.

Apple's philosophy of a simple, elegant and straightforward interface remains, thus preserving the user experience that made the Mac famous. However, they've added some new additions that are sure to impress. The Apple Menu and desktop still remain in their familiar guise, but within the Utilities folder contained on the hard drive can be found a slew of eye-popping new gadgets and utilities designed to offer the power user access to the nuts and bolts of the operating system. For those of you who have ever used other Unix-based operating systems, such as SGI's Irix or the many flavors of Linux, some of these tools will look very familiar to you. There is now the ProcessViewer, which will allow you to view and control all running applications and hidden processes on the Mac OS. One of the most important of these functions is the "force quit," allowing you to unconditionally terminate an application without affecting system integrity. Another useful utility, and one of my personal favorites for usefulness and the sheer humor factor, is the Terminal application. Yes, Virginia, the Mac now has a command line!



How is a command line interface useful, you ask? After all, one of the cornerstones of the Mac legacy is the graphical interface, eliminating the need for users to memorize awkward commands for basic tasks. I'm not referring to the command line as the end-all and be-all of Mac OS evolution, but rather, as what it represents. OS X is all about granting power level by level, like layers of an onion; if you need access to the more complicated aspects of system management and development, you have it. With this array of new, more powerful tools, the Mac begins to offer more things for more people, meaning the possibility of an increased

user base. From the perspective of a network administrator or software developer, the command line represents an important addition to the functionality of the Mac for high-end users. Things such as OS X's built-in networking protocols, like FTP or Telnet, can be accessed directly through the Terminal interface. Because of the underlying Unix architecture, users can have direct access to remote systems, which opens up new possibilities for render farms. Render farm networks can now be controlled from a remote location, including starting a render node, processing a scene, monitoring the remote system's progress, and even powering down the system when the job is finished!


Process Viewer Processes Edit Window
Thu 2:37 PM
Sylia's Terminal — /bin/tcsh (ttyp2)
Processes: 48 total, 3 running, 45 sleeping... 134 threads 14:37:11
Load Avg: 0.20, 0.18, 0.18 CPU usage: 17.7% user, 9.5% sys, 72.8% id
SharedLibs: num = 96, resident = 20.2M code, 1.36M data, 4.74M LinkEdit
MemRegions: num = 3285, resident = 189M + 9.06M private, 113M shared
PhysMem: 75.5M wired, 50.2M active, 387M inactive, 512M used, 512M fre
VM: 2.95G + 44.2M 11638(0) pageins, 0(0) pageouts
PID COMMAND %CPU TIME #TH #PRTS #MREGS RPRVT RSHRD RSIZE VSI
628 top 13.0% 27:29.64 1 19 14 188K 224K 408K 1.3
623 tcsh 0.0% 0:00.18 1 17 14 248K 460K 688K 5.5
614 Modeler 0.0% 0:10.73 5 107 212 18.5M 22.4M 25.8M 11
604 Hub 0.0% 0:02.34 3 81 86 2.16M 6.80M 4.92M 76.
347 System Pre 0.0% 0:01.29 1 66 60 1.84M 5.31M 4.20M 76.
342 Grab 32.0% 0:16.95 3 132 104 11.0M+ 22.6M 14.3M+ 87.
Saber_1
Process Listing
Find:
Show: All Processes
Name User Status % CPU % Memory
inetd root Running 0.0 0.0
init root Running 0.0 0.0
ipconfigd root Running 0.0 0.0
kextd root Running 0.0 0.1
LaunchCFMApp cailean Running 0.0 2.5
LaunchCFMApp cailean Running 0.0 3.3
cailean Running
cailean Running 0.0
root Running 0.0
root Running 1.8
root Running 0.0
root Running 0.0
root Running 0.0
Process ID Statistics
614
Parent Process ID: 54 (Window Manager)
Process Group ID: 54
501 (cailean)
Apr 19 10:55:40 localhost WindowServer[54]: CGSOrderWindow: Invalid other

OS X offers the ability to run your legacy applications seamlessly under OS X in an emulated version of OS 9.1, dubbed the Classic Environment. The environment is very well done, and when activated, actually comes up with a separate collapsible window showing the 9.1 startup sequence, extension icons and all. OS 9.1 actually boots faster under the Classic Environment, and does so without competing with other applications for system resources. Unfortunately, there's a price to pay with the emulated operating system: due to the way Classic functions under OS X, it is cut off from directly accessing your hardware. This can limit OS 9.1 applications in aggravating ways: for example, Wacom tablets no longer respond in Photoshop under OS X. Most users will soon find themselves tracking down the Startup Disk control panel and rebooting into 9.1 in order to get any serious work done.

The most crucial issue for OS X is going to be acceptance by third party vendors, in the form of application and driver support. No matter how promising and exciting a new platform is, it should not be assumed that third-party vendor support is going to automatically follow. Vendors must carefully weigh the wisdom of such an investment of time and manpower before launching full-scale into such an undertaking. As I mentioned before, it isn't just the interface that's changed; the old core architecture is a thing of the past. The new engine under the hood is the Mach micro kernel, and it comes with its own set of rules.

Apple has done its best to make sure this transition goes as smoothly as possible for its vendors. The Apple team had the foresight to design and implement the so-called Carbon standard early on in the game, giving program developers a chance to start creating OS X-compliant software even before the OS itself began to take shape. Apple has also added to this with the new Cocoa API, an object-oriented coding interface that is supposed to allow Mac programmers to create code more easily and with fewer headaches than ever before. What does this mean for Mac users? A friendlier system means an easier task for programmers, who can enjoy not only the new Cocoa environment for programming tasks, but also built-in Perl, Java, and of course, AppleScript. All of this will encourage existing vendors to continue supporting the Mac OS. More importantly, it encourages new vendors to hop on board.

Speaking of which, everybody's favorite company, NewTek, has stumped 'em all and released LightWave 3D version 6.5b for OS X with lightning speed, well within the magical 60 days following OS X's release. Since when is a software company early, let alone punctual? (The rumor mill predicted a full release no earlier than the end of May). Let's hope this marks a continuation of the good things coming out of the NewTek brain trust in San Francisco. The new release includes a host of improvements, with render time, as well as render quality greatly improved (think "useable radiosity"). NewTek has added improvements in its character animation software, and a multitude of subtle refinements to the overall

user experience.

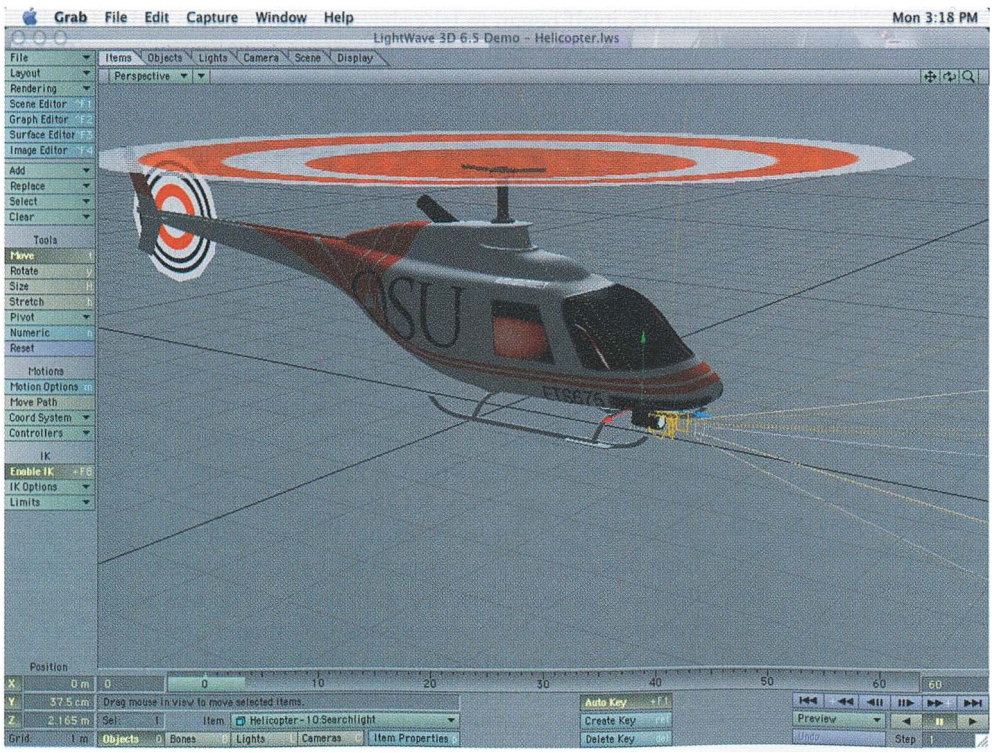
As of this writing, many of the most popular software vendors in animation, graphic design, and video editing are still in the development stages of creating OS X-native applications. However, each of these companies has pledged continued support of the Mac platform, and they are working closely with Apple engineers to see to it that their users are given solid new products to work with OS X as soon as possible. Here is a list of these notable names, and their plans for an OS X release:

* **pmG**: Makers of the extraordinary character animation software known as project:messiah, pmG (headed up by Dan Milling, Lyle Milton, and Fori Owurowa) is in development of an OS X version of their product's next exciting incarnation: messiah:studio.
* **Electric Image**: Universe 3.0, the latest installment of the original 3D application for the Mac, is close to a full release of an OS X-native version of its new software.
* **Maxon Computer**: Cinema 4D XL and BodyPaint 3D are in public beta as well. No word yet as to an official release date.
* **Alias|Wavefront**: A|W's much-anticipated release of Maya for OS X is in the works, with a release slated for the third quarter of 2001. The initial release for OS X is will be the all-new Maya 4.0.
* **Adobe**: Adobe has stated that there will be no patches or upgrades to include OS X-native support to any of their applications until the next major releases of their popular line of desktop graphics software.
* **Media 100**: Expect to see the next version of Media100i, the popular nonlinear editing system, by the end of this year. Dedicated to rock-solid support and positive user experience, Medial 00 is working closely with Apple and its third-party partners to make sure that no detail is left unfinished.
* **Thursby Software**: Makers of the eloquent Mac/Windows networking solution known as DAVE, Thursby has already demoed a working version of this popular Mac application at MacWorld San Francisco. They plan on having a release version by MacWorld New York in July.

Apple has chosen a new direction, and that direction will be OS X. Although the new operating system heralds the end of the Mac's days of underpowered operating systems, this is only the first step. I hate to break it to you, but OS X isn't perfect. Despite its promise of rock-solid stability and workstation-grade system design, Apple still has some bugs to iron out. Since I started running OS X a month ago, I've actually managed to lock up the entire system twice (granted that that's twice in one month, which is still a pretty good statistic). OS X, despite the name, is a version 1.0 operating system.

If you're a software developer, a web site administrator, or just a Mac enthusiast, then I would definitely encourage you to buy yourself a copy of OS X and to begin familiarizing yourself with it. As for the animators, artists, and editors out there: For the majority of the creative types out there, I recommend sticking with OS 9.1 for now. Just like many things worth waiting for, OS X is going to take time. Users and developers will need to put OS X through its paces and work out the kinks before it can be seriously considered for any kind of production pipeline.

As Apple and its partners continue improving and developing OS X, we will see an assortment of high-end application being released in the next six months, with widespread support being offered within the year. Does OS X herald a change in the winds of the overwhelmingly Windows-centric PC market? Will Windows users flock in droves to buy brand-new Mac OS systems for every home and office? Probably not. As far as this Mac user is concerned, the important thing is that the Mac is still going strong, and the future looks bright. Everything else is just details.



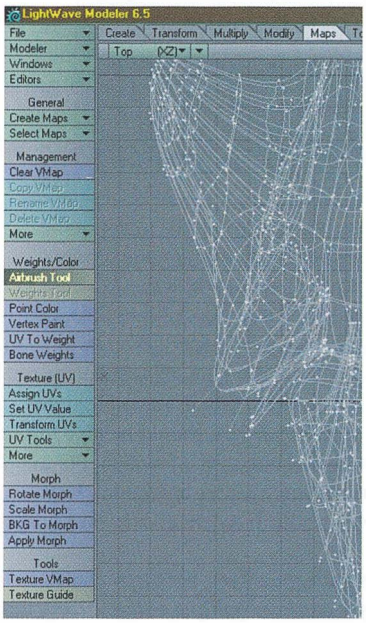
Cailean Babcock has worked in computer animation for over five years, spending the last three at Educational Television Services at Oklahoma State University. He also moonlights in the wild and wacky world of freelance animation, with work featured on the Venture Outdoors series on Fox Sports, as well as his own character animation projects. His email address is cailean@provalue.net.

Texturing with Vertex Maps

by Barrett Meeker

Vertex maps, new to LightWave 6.0 and improved in 6.5, are simply another set of data assigned to a model's vertices. Just as each pixel of an image has its RGB color values and each vertex of a 3D model has its XYZ location in 3D space, each point of a vertex map has a value assigned to it. When you create a new vertex map you are overlaying another set of data on an object, giving each vertex of that object an additional value. There are several different types of data that can be applied using vertex maps.

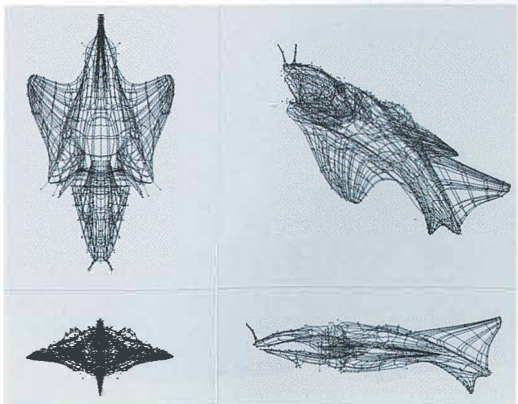
A weight map assigns a single value, usually shown as a percentage. Weight maps are commonly used to control the influence of a skeleton's bones. You can see the weight map's values represented on the model by selecting Weight Shade from a viewport's title drop-down menu. A vertex color map assigns a color to each vertex of an object. You can only see this effect after applying the vertex color map to the surface. A UV map assigns U and V coordinates. To see this graphed in LightWave, select UV Texture from a viewport's title drop-down menu.



Before we get into this tutorial, I recommend that you either edit your current modeler menus to include a Maps tab, or download my config files from **www.gluge.com/LWvtxTutorial.exe**. This self-extracting archive will decompress into several configuration, image and model files. If you choose to create your own Maps tab you should at least include these tools: Create Vertex Color Map, Airbrush Tool, Weights Tool, and Point Color.

Adding Color with Weight and Color Maps

The LWvtxTutorial.exe file mentioned previously includes the model you see in this tutorial's figures. You can use this object, or choose one of your own. This technique works best with a smooth organic model rather than a mechanical or sharp-edged object. Make sure the object you use has an adequate number of vertices, somewhere in the 1000-2000 range.

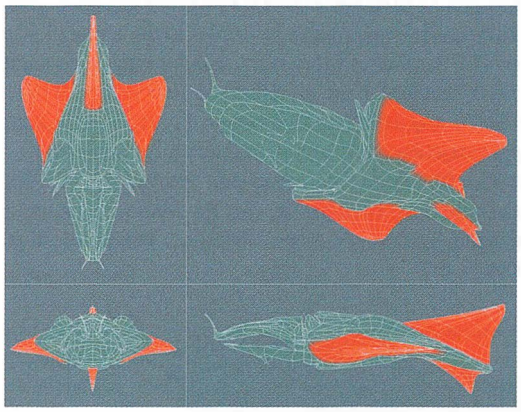


Load the object. Select some polygons or points you wish to color, such as the fins of the fish. Remember that if you are selecting polygons, all of each polygon's points will have the value assigned.

At the bottom right of the screen, there are three buttons, W T M, for Weight map, Texture map, and Morph map (or Endomorph). Highlight the 'W' by clicking on it. From the pull-down menu to the right of the buttons, select (new). This will bring up a Create Weight Map selector. Type in a descriptive name such as "fins", "ears", or whatever. Leave Initial Value checked, and the value at 100%. Click OK.

This is creating a weight map, and giving the polygons or points you selected a value of 100% and everything else a value of 0%. If you have nothing selected and you leave Initial Value checked and the value at 100%, everything will be affected (if nothing is selected, everything's affected). If you want to make a blank weight map, simply uncheck Initial Value when you create the weight map.

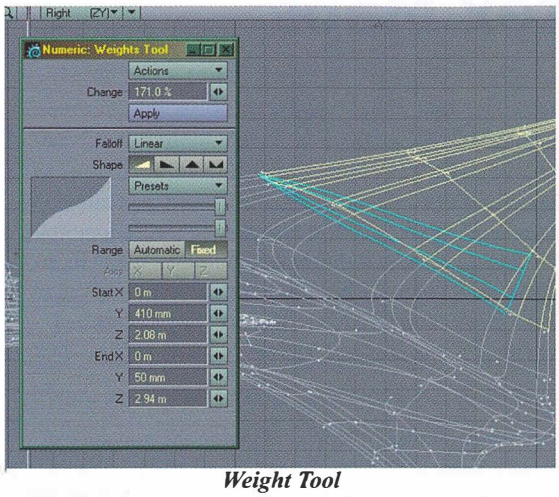
To see your newly created weight map, select Weight Shade from a viewport's title drop-down menu. If you have applied more than one weight map, you can choose which weight map to view with the drop-down menu in the lower right hand corner. For weight maps, the color red represents a positive value, blue is negative and green is neutral. For the purposes of keeping things simple in this tutorial, we will be making everything shaded green one color and everything shaded red another color.



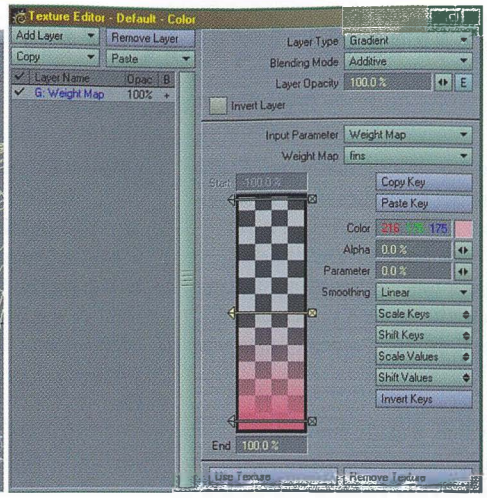
I recommend that when making vertex maps, you should follow the natural contour lines of the object. This will help make the resulting textures more believable.

You can adjust your weight map in two ways. You can select the polygons or points you want to change, then select Weight Tool from your Maps tab. Press 'n' to bring up the numeric options for this tool. The falloff pull-down gives you some great control. For example, select Linear, then right-click-and-drag in one of the viewports to make the falloff guide appear. This lets you control the bounds of the falloff. The Change value is the amount of change that will be applied; it's an additive effect, but you can enter negative values.

The other tool is perhaps more fun, but less useful. You use the Airbrush Tool (called Weights Brush in 6.0) by selecting it, then dragging the brush over the vertices of your object. This will increase the weight value of the vertices; to decrease, hold the Shift key down



Weight Tool

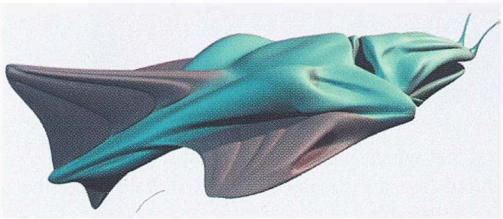


Weight Gradient

when left-clicking. To adjust the size of the airbrush, right-click-and-drag.

Now that you have your beautiful weight map, save your object, and send it to Layout. Open the Surface Editor, find the surface of your object, and click the T button next to Color. This will open the Texture Editor. In the Texture Editor, change the Layer Type to Gradient. The goal here is to set up a gradient what will take values from our weight map, and assign a color based on those values. From the Input Parameter pull-down select Weight Map, and select the weight map you created.

Click anywhere on the gradient to add a key, a point on the gradient that has its own color. Add a few more keys and give them different colors. The Alpha value is similar to the alpha of a layer in Photoshop. The Parameter value is the exact weight map value that corresponds to that key. Note the Start and End values of the gradient: these are -100% and 100%, and correspond to the weight map values. Set a key at (parameter) 0% and give it an alpha value of 0%. Give the key at -100% a 0% alpha value also. Add another key at 100% and assign it the color you want for the fins (or whatever you colored red when you made your weight map). Make sure the alpha value for this key is 100%.



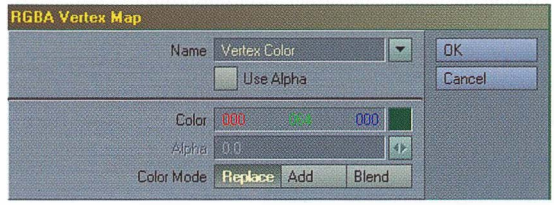
Click Use Texture and do a test render. Your results should look like this.

This is a good place for you to go back and experiment. Try using negative weight map values, move the keys around, try whatever comes to mind until you are comfortable with what's going on and feel that you understand how it works. Create some more weight maps, and add some more layers to the color texture until you're happy with the coloring. Vary the kinds of weight maps; try making some randomly with the airbrush to help achieve that organic look. Use the alpha values to control what shows through from the bottom layer.

Coloring with Color Vertex Maps

An alternative way of using vertex maps for color texturing is to apply a vertex color map, a new feature in LightWave 6.5. Create a vertex color map, at the menu put in whatever name you like, and uncheck Use Alpha. Use Alpha means your vertex map will have both a color value and an alpha value for each point, which means the base color will be able to show depending on the alpha value. Wh this can be useful, it can also get confusing, so I suggest that uncheck it for your first time and back and experiment with it

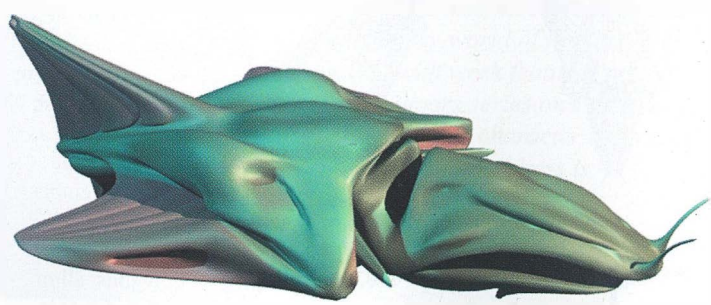
Now that you have your map created, select some points or polygons which colors you wish to change, just as we did with the weight maps. Now instead of using the Weight Map tool, select Point Color. Here uncheck Use Alpha also, for the same reasons as mentioned previously. Select a color and a Color Mode; Replace is usually the mode you want. Press OK.



Nothing appears to happen, because you have to apply the map to the object before you can see the effects of the vertex map. Open the Surface Editor, select the surface, and go into the Advanced tab. About halfway down you should see Vertex Color Map. Select the vertex color map you made, and ta da, you should see it. Vertex Coloring is just the opacity percentage for the vertex map. If you're not seeing anything, be sure your preview render mode is set to 'Texture' (You can change it in the viewport title bars.). You should be able to see the vertex map in Layout as well, which is one of the few benefits to using Vertex Color maps. After playing with it for a few hours, I've found the Point Color Tool to be very weird. It rarely does what I want.

While Vertex Color maps have the advantages that you can see the colors in the preview and you can use multiple colors, I very much prefer Weight maps. You can use weight maps for bump, specular, and other attributes, not just color. You can also use them to control procedurals, as you will see in a bit. You can easily layer your weight maps in whatever channel you are applying them; if you use a vertex map, you only get one. With weight maps you can easily change your colors from the surfacing panel, rather than having to go into modeler. But Vertex maps can be fun.

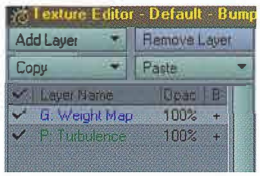
For the final coloring I used 6 weight maps.



Controlling Procedural Textures with Weight Maps

Now let's put a procedural bump map on this sucker to help break up the specular highlights and give some feeling of texture. Open up Layout, load the object, open up the Surface Editor, and find the surface. Click on the T button across from Bump to add a texture layer on the bump channel. Change this texture layer type from Image to Gradient, change the Input Parameter to Weight Map, and select a weight map.

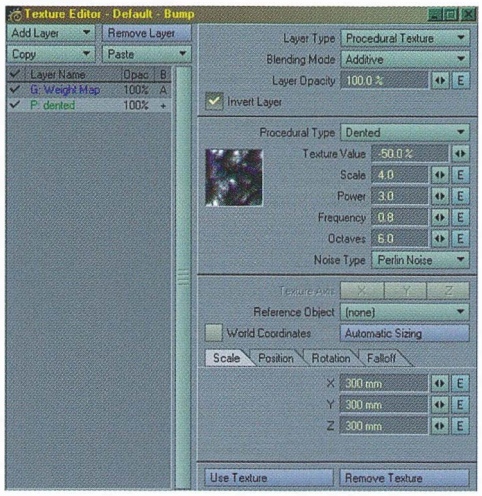
Add a layer, in the top left corner of the Texture Editor, select Procedural from the Add Layer pop-up. You will now see two texture layers; if you have 6.5 they should be nicely color-coded for you with the procedural in green, the gradient in blue, and image layers appearing in red. The procedural we just added should appear above the gradient; we need to change that, so click and drag the procedural layer so it is under the gradient layer.



The next step is to set up the gradient as an alpha map, to fade the procedural in and out. Go to the gradient layer and change the Blending Mode (right under Layer Type at the top) to Alpha. This means the gradient works like an alpha map for the layer directly under it; when the gradient value is 0% (black) the layer directly under the gradient will not appear, and when the gradient value is 100% (white) the layer directly under the gradient will show though. The gray values in between will let the layer below show through proportionately.

Now you are ready to add some keys to that gradient. I used the following keys: (Parameter -100%, Value 0%), (Parameter 0%, Value 0%), (Parameter 30%, Value 100%). When the weight map is green or blue, the bump procedural won't appear; but as soon as the weight map starts to get red, the bump map appears pretty fast, because at 30% the gradient value is already 100%. I did this mostly because I painted my weight map rather lightly in spots, but I still wanted to be able to notice the bumpiness in those spots. When you're happy with your alpha gradient, select the procedural below it to bring up its properties.

Usually I decide what kind of look I'm trying to get, and from that decide on what Procedural Type to use. If I'm making a nebula cloud, I'll use a Turbulence or Fractal Noise, and Crumble can work well for rock. I highly recommend simply going through them all and trying them out. When you get some idea of which procedural type you are using, you can start the tweaking process. I usually play with the scale of the texture, a parameter which all Procedural Types have. Scale is very important to the results you get. I'd suggest you start with a value between 1/5th to 1/15th of the size of your object, using that value for the X, Y and Z scales, then go through and change the other parameters. These are the values I settled on.

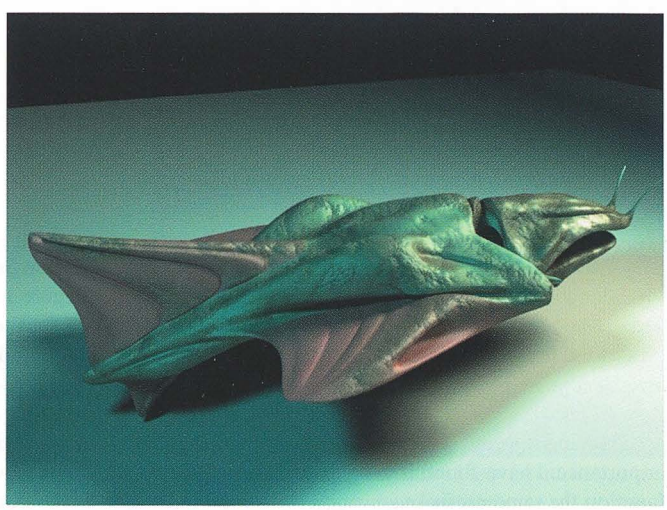


Other Interesting and Useful Tools

Vertex Paint is a free plug-in that you can find through flay.com. This is a great plug-in for painting Vertex Color Maps, and it is much better than the standard tools. It has its own real-time preview window, and it can be used for Vertex Color Maps and Weight Maps. But the best feature doesn't really apply to texturing: you can create a skelegon for your object, make weight maps for the skelegons, then run Vertex Paint. This gives you the ability to both deform your mesh with the skelegon you made and change the weight map interactively to see how it's affecting the deformation.

I found the technique of using weight maps to blend UV image maps invaluable in texturing complex objects with image maps very precisely. However, it's a very tedious process. You select some polygons, create both a UV map and a weight map for them, then paint the UV map, then apply it using the weight map to blend the UV with the surrounding texture.

*You can email Barrett Meeker at **gluge@gluge.com**, or see more of his artwork at **http://www.gluge.com**.*



Advanced Character Setup Using Sock Monkey

by James Jacobs

When I create a character rig, especially one that is to be used by many animators, I continually consider the following list of questions while setting up the control structure:

1. Is it easy to select and articulate only those controls that the animator is interested in?
2. Do the controls make visual sense?
3. How can I reduce the number of control nodes to improve workflow?
4. Is there enough low level control of the character to allow the animator to act?
5. Has the setup been 'padded' enough to allow extra low level control to be added during production if necessary?

Having considered these points and consulted with the animators, I then go off and attempt to create a puppet that will meet the supplied criteria.

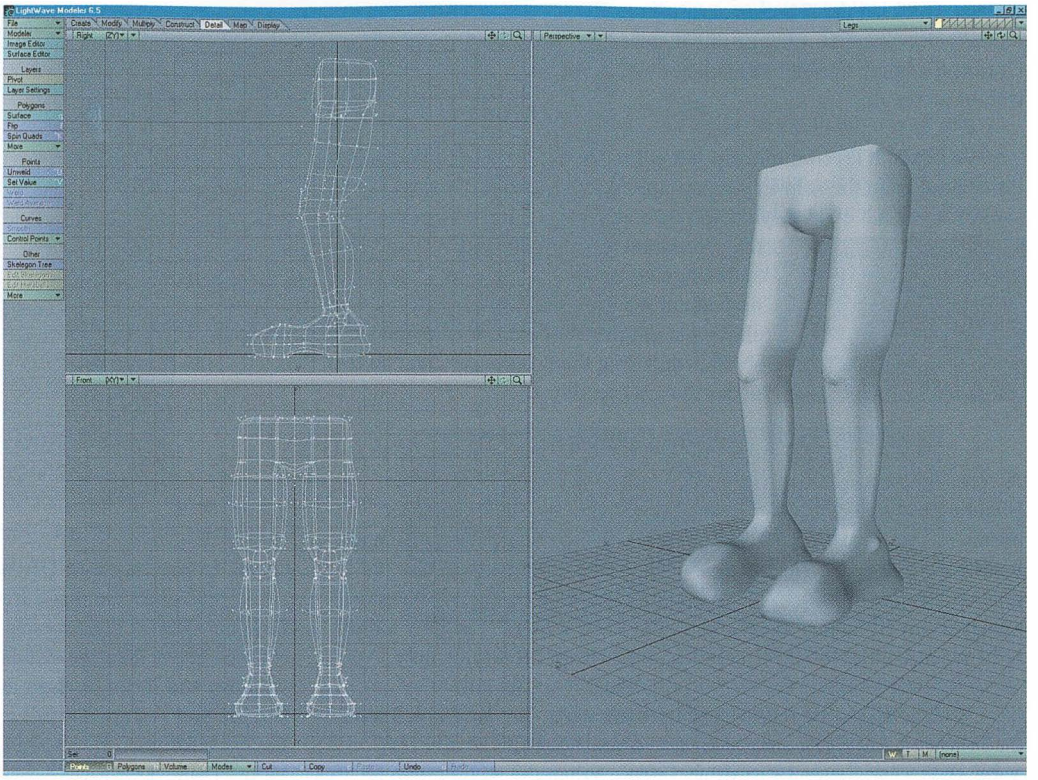


Figure 1: *Modeled leg geometry*

When NewTek upgraded LightWave to version 6.5, they added a new deformation tool, SockMonkey, to its arsenal. SockMonkey can be found in Object Properties under the Deformations tab, below Add Displacement. This plug-in is LightWave's implementation of the point clustering paradigm that exists in other applications.

SockMonkey enables you to define cluster membership by using either weight maps or point groups. In either case, membership of each point is exclusive to only one SockMonkey Control Item. That is, if a point is included in more than one Vertex Group or Weight map, that point will only be affected by the first deforming group that SockMonkey encounters. It is also important to note that vertices that have been included in a SockMonkey Group cannot be affected by other deformation types. Consequently, after you set up the initial character rig, you must either rework the point membership so as to not include localized areas that require further deformations, or subdivide the clusters into smaller point groups that when parented together move as one but also have the ability to move independently.

If you haven't already downloaded the DS_CenterPivot plug-in, do so now. It will help make the process of following this tutorial easier. You can find the plug-in (along with many other useful tools) at:
http://www.dstorm.ab.psiweb.com/dslib/index.html

Step 1: Modeling the Legs

We will be creating this character setup using Point Selection Sets. Open Modeler and model a pair of SubPatch legs facing down **Z**, Figure 1. Be sure to add enough detail in the areas that will deform. When you are finished, store the legs in the first layer of your object file. Name your object Legs, and your first layer gLegs.

Having rigorous naming conventions in a character setup is of the utmost importance. I have found it to be extremely useful. I give all nodes with a similar function the same prefix, e.g., geometry is prefixed with a lower case g, joints

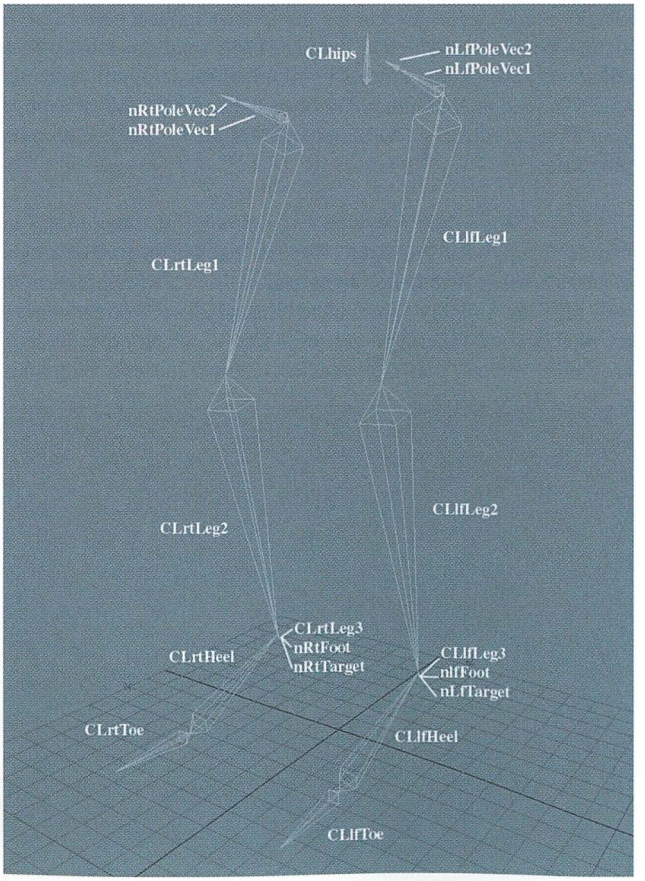


Figure 2: *Skelegon placement with names*

with a lower case j, and controls are named using all caps. Having consistent node names enables you to write expressions with ease, and to write programs or LScripts that take advantage of known naming conventions.

Step 2: Creating a Skelegon Tree

Create a Skelegon in layer two using the hierarchy illustrated in Figures 2 and 3. Name the second layer nLegControlBones (n referring to Null).

Step 3: Defining Point Groups

Following Figure 4, create Point Selection Groups for each of the illustrated regions. Be sure to name these groups appropriately, as the SockMonkey plug-in will later reference these group names to speed up the rigging process.

Step 4: Building the Controls

It is my habit to create geometry to be used for controls, rather than to rely solely on Null objects. The purpose of this is to clarify the function of the controls by creating icons that can be selected in an interactive view.

Create a new object in Modeler. Open the layer view. Select the first layer of the new object as your foreground, and the nLegControlBones layer from the Legs model as your background layer. Create a box object that is elongated from top to bottom, centered around the base of the nLfFoot bone. Kill the polygons, and reconnect the points with two-point polys. Swap layers, select the vertex at the base of nLfFoot and copy it. Swap layers again and paste the point into the current layer. Select the vertex by itself, and execute DS_CenterPivot. This will cause the pivot to be centered inside of the selected point. Save the current object as LF_FOOT.

Create a new Object. On its first layer, create a disc that lies flat on the XZ plane. Again, select nLegControlBones as your visible background layer. Move the disc so that it is located underneath the CLlfHeel bone that is located in the background layer. Kill the polys, and reconnect the points with two-point polygons. Select the three points on the disc that are closest to the back of the object. Shape those points so that they form a vertical triangular extrusion. This will allow the disc icon to be seen from all orthogonal views. Again, swap layers. Select the vertex at the base of CLlfHeel and copy it. Swap layers again and paste the point into the current layer. Select the vertex by itself, and execute DS_CenterPivot. Save the current object as LF_HEEL.

Copy LF_HEEL's geometry. Create a new object. Select nLegControlBones as your visible background layer. Paste the geometry to the first layer of your new object. Move the geometry so that it is centered underneath CLlfToe, and move its pivot to the appropriate location. Save the current object as LF_TOE.

Duplicate the above steps for the right side of the setup. I recommend that you rotate the right side's heel and toe controls 180 degrees around their respective centers. This will allow for easy differentiation between the left and the right controls from a side orthogonal view.

Create a cube object titled BLOCK, with its pivot placed at its center. Create another that is stretched along the X axis entitled HIPS. Center HIPS around CLhips (that is again located in nLegControlBones), and copy its pivot to the control.

Finally, to create the pole vector controls (controls that will determine the orientation of the plane that the IK is being solved along), create an arrowhead shape out of two-point polys. Place it in front of the leg you wish it to affect. Select its points and center its pivot on your selection. Do the same for the other leg, saving each of them as lfPoleVec and rtPoleVec respectively.

When you are finished, your controls should resemble Figure 5. Make sure that your pivots are in the same locations as those indicated in the image.

Step 5: Bringing It Into Layout

Now that everything is prepped and ready, the fun part can begin. Start Layout and load the Legs object. Open the Plug-in Options window. Under the Generics drop down list, select Skelegons2Nulls. Set the Object to Convert to Legs:nLegControlBones. Toggle Apply Skelegon Hierarchy off, and Place Near Object Anyway on. Choose OK.

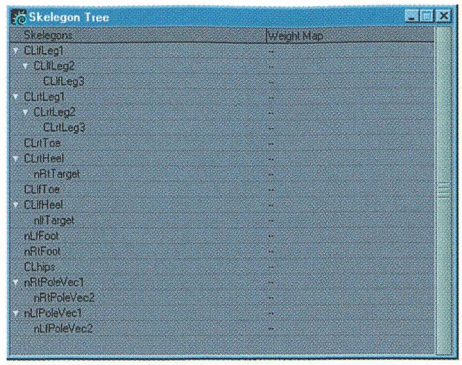


Figure 3: *Skelegon Hierarchy*

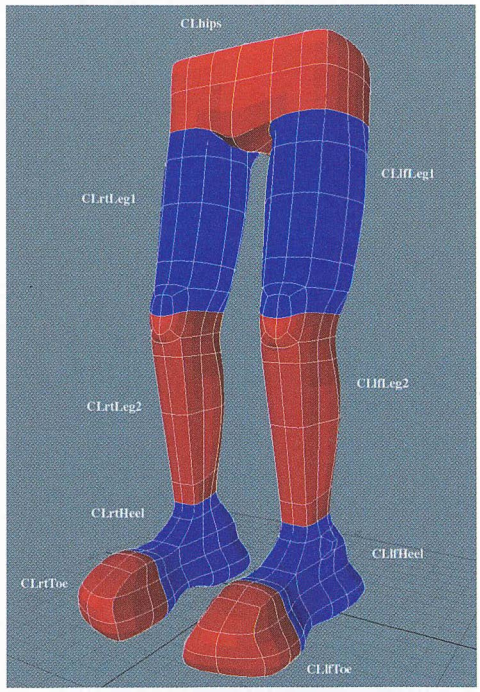


Figure 4: *Colored areas represent different Point Selection Groups*

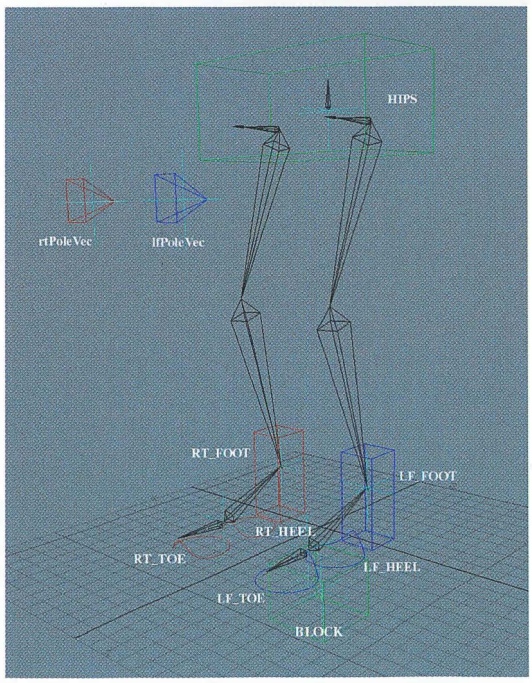


Figure 5: *Modeled controls, their locations, and their pivot locations*

Select Legs:gLegs and open its Object Properties. Under the Geometry tab, set its Subdivision Order to Last. Under the Deformations tab, choose LW_SockMonkey from the Add Displacement drop down menu. Double click on LW_SockMonkey to reveal its interface. Select Auto Assoc. w/Existing.

After the cluster relationships have been made, select each group listed at the top of the SockMonkey window individually. While selected, toggle off their bounding box display. When finished, select Continue.

Make sure that Parent in Place mode is active (found in General Options), and parent your items until your Schematic view resembles the one pictured in Figure 6.

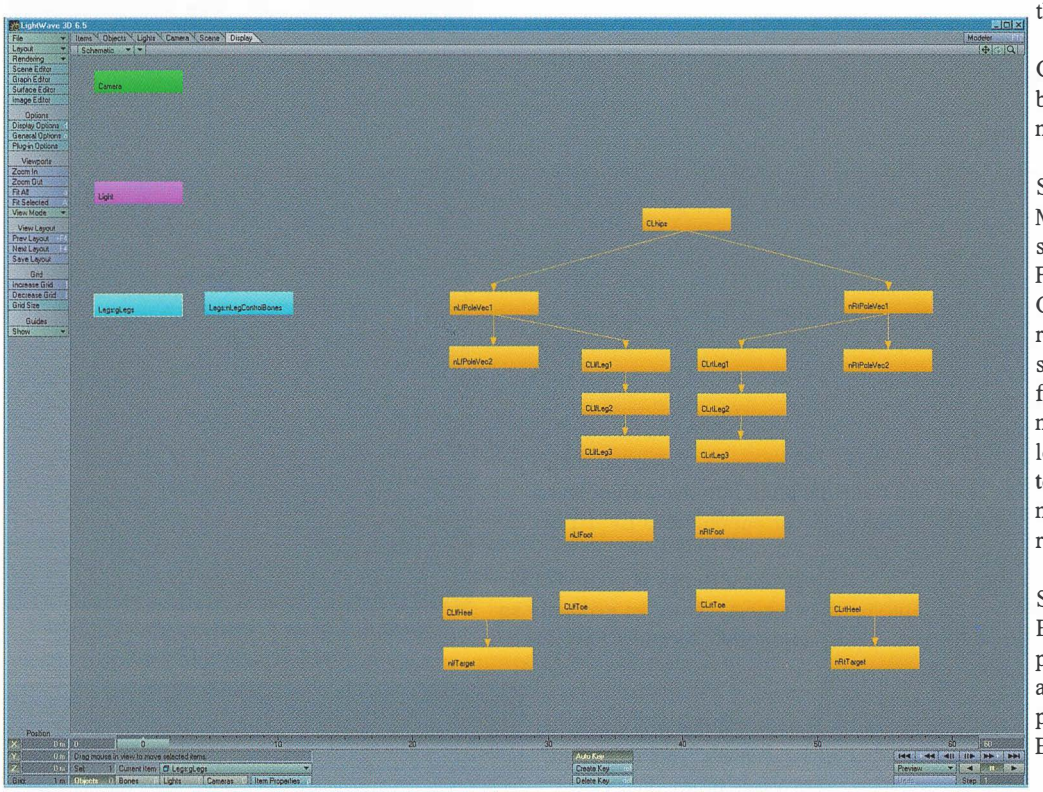


Figure 6: *Schematic view of the Sock Monkey nodes in hierarchy*

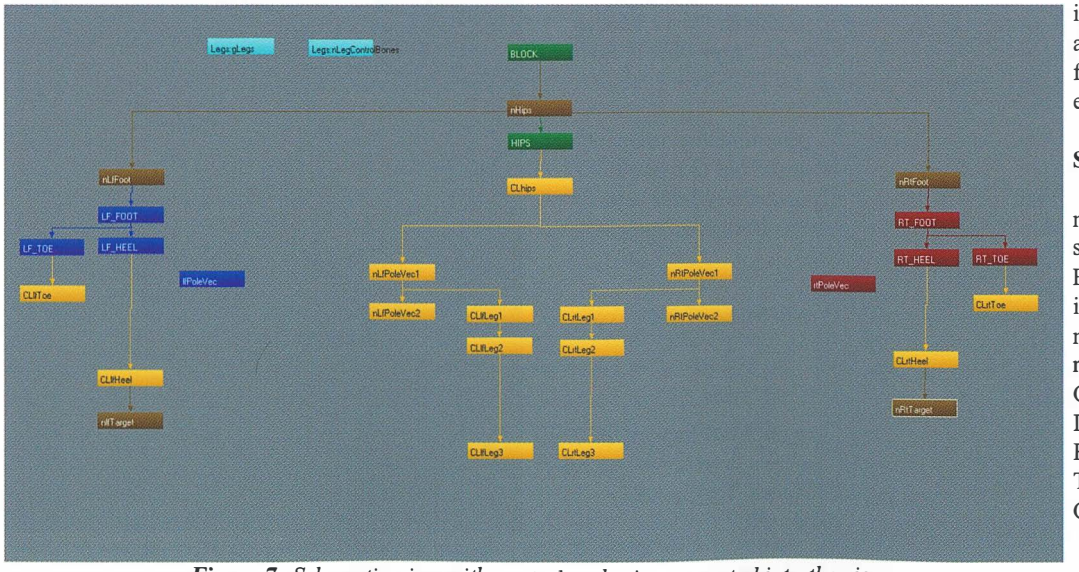


Figure 7: *Schematic view with control nodes incorporated into the rig*

Load the control objects. Set a rest pose keyframe for all items and all channels at frame 0. Continue to parent items until your Schematic view resembles that pictured in Figure 7.

Before we continue, allow me to explain the workings of this rig. The items pictured in Figure 7 that are colored either red, green or dark blue, are the control nodes. The items that are colored orange are the cluster handles, the brown nodes are null objects, and the cyan items are source geometry. The red control nodes operate the right side of the character. Blue controls the left, and green articulates the middle. The function of the BLOCK node is to move the entire character rig around. The HIPS node articulates the character from the hips. The LF or RT_FOOT move or rotate the respective foot. The LF or RT_HEEL allows each foot to lift its heel off the ground. The LF or RT_TOE manipulates toes. Finally, the lf or rtPoleVec controls the placement of the character's knees.

Create five Null objects. Label them: baseNode, nOutsideHiearchy, nBlock, nLfPoleVec and nRtPoleVec.

Select nLfPoleVec and open up its Motion Options. Under Add Modifier, select LW_Follower. As its Item To Follow, choose lfPoleVec and press Continue. When the window closes, remove Follower from nLfPoleVec, and set a keyframe for all of its channels at frame 0. (What this will do, is move nLfPoleVec so that it is in the same location as lfPoleVec.) Parent nLfPoleVec to LF_FOOT, and parent lfPoleVec to nLfPoleVec. Repeat the process for the right side.

Select nBlock and make it follow BLOCK. Remove LW_Follower and parent BLOCK to nBlock, parent nBlock and nOutsideHiearchy to baseNode, and parent Legs:gLegs and Legs:LegControlBones to nOutsideHiearchy.

Select BLOCK, lfPoleVec and rtPoleVec individually, and if they are not already zeroed, set all of their transformations to zero. Set a keyframe on everything at frame 0.

Step 6: Setting Up the IK

Motion Options for nodes nLfPoleVec1 and nRtPoleVec1 are set to IK on Heading, Pitch and Bank. nLfPoleVec2 has lfPoleVec as its Full-time IK target, while nRtPoleVec2 does the same with rtPoleVec. CLlfLeg1, CLlfLeg2, CLrtLegl and CLrtLeg2 are all set to IK on Pitch and to keyframe on Heading and Bank. CLlfLeg3's Full-Time IK target is nLfTarget, while CLrtLeg3's is nRtTarget.

At this point, your setup should

resemble Figure 8. All that is left to do is to lock everything from selection except the control nodes, hide everything from view except the geometry and the control nodes, hide the control nodes from the renderer, and lock off the following channels from control nodes that you'll want to restrict:

LF_TOE - position X, Y, Z
RT_TOE - position X, Y, Z
LF_HEEL - position X, Y, Z
RT_HEEL - position X, Y, Z
lfPoleVec - rotation H, P, B
rtPoleVec - rotation H, P, B

To hide the control nodes from the renderer, set the Object Properties for each node as follows: Unseen by Rays, Unseen by Camera and Unaffected by Fog should be on, and Self Shadow, Cast Shadow and Receive Shadow should be off.

Figure 8: *Schematic view with nulls added for the purpose of zeroing out the transformations of select control nodes. Ik dependency is also shown.*

One of the nicest features of SockMonkey is its scalability. We could now continue adding geometry to the source mesh and the selection groups, and weight maps would grow to include the new geometry. This is especially useful if your character is in development, where revisions abound.

As far as character rigging is concerned, this is just a beginning. I look forward to traveling farther down this road with you in the future.

James Jacobs is a Senior Animator at Toybox, a visual effects and post production facility located in Toronto Canada. He has received an Emmy and a Gemini Award for his contribution to the production of Nelvana's Rolie Polie Olie, *and has been recognized in* Animation Magazine *for his work on a series of Hershey Kiss spots produced at Guru Animation Studio. James has filled such diverse roles as digital cel painter, compositor, render wrangler, technical director, overseas technical director (in Taiwan), senior technical director, and senior animator.*

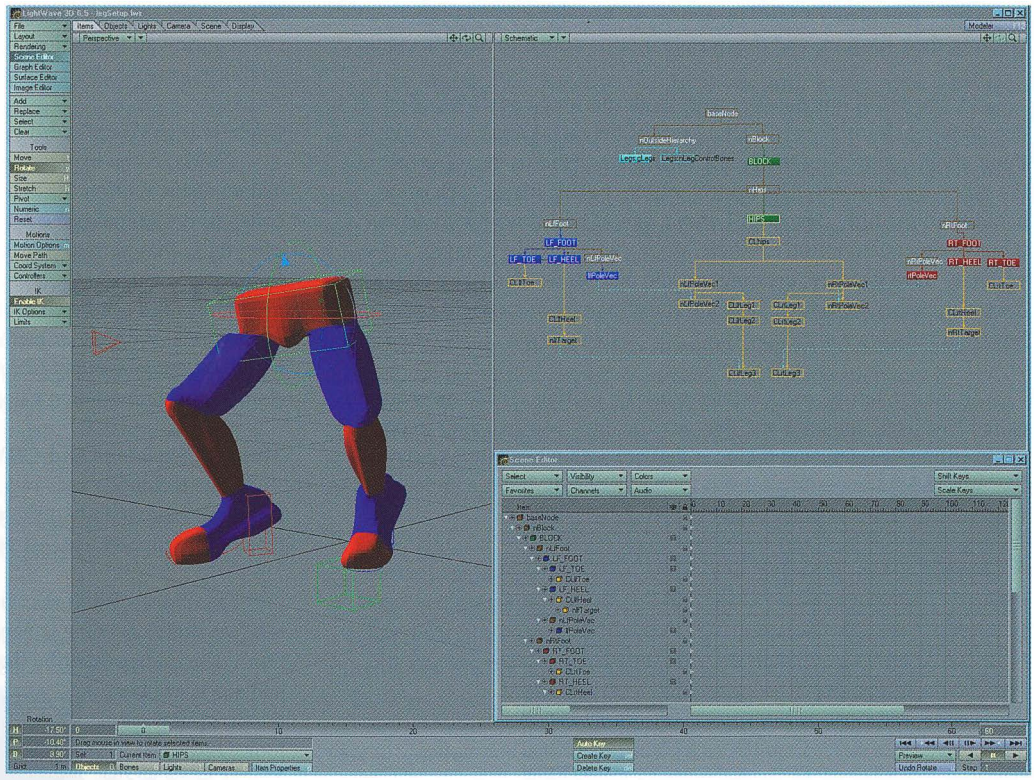


Figure 9: *Final character rig*

Continued from Page 32

visual effects to show the scope and intensity of combat and the spectacle of civilizations long past will become increasingly commonplace. We as effects artists will surely benefit, but we won't be alone. The movie-going public will reap the rewards as filmmakers stretch their horizons and open up to the possibilities, which I can say without hyperbole are truly becoming unlimited.

Considering the leaps we've seen just in the past five years, it's obvious that the next five are going to give us no shortage of visual treats. Of course, the fact that soon all of these effects will be done on Amigas just makes it all the more interesting. No, really, I saw it in my crystal ball...

Steve Pugh is the General Manager at Foundation Imaging. You can contact him through its website, www.foundation-i.com.

Messiah Muscle, Part I

by Ron Wolfe

Muscle motion is the movement of muscle and skin. Muscles come into play with computer animation when you are creating your own "illusion of life". People, animals, and even our perception of mythical and fantasy creatures use muscles. We are accustomed to the way real things move, and though the subtle movements of muscles and skin can be taken for granted, they can often make the difference between what is believable and what is not.

Many can still remember the awe felt when first watching *Jurassic Park*. Here were living, breathing dinosaurs which looked like they meant serious business. Even though *Jurassic Park* was a breakthrough in computer animation, it was a scene from *The Lost World* that I remember for realistic muscles and skin. It was early in the movie when those lumbering Stegosaurus walk across the screen. I could see the skin bulging and stretching, sagging and lifting, and rolling as huge legs plodded ahead. In my mind, that Stegosaurus had "weight" - and plenty of it. The animators had done a great job creating their illusion, and the muscle motion added to its believability.

In you and I, our muscles work with minimal conscious effort. If I decide to reach for my sixth Diet Dew of the day, I don't consciously think about contracting my bicep, raising my hand or the movement of the other arm muscles. The control system buried inside each of us handles all the little details as I grab a can of pop. I wish I could say there is a messiah option to "Activate realistic muscle behavior". As you may have guessed, creating muscle motion for computer animation does not happen automatically, and it causes you to stop and consider how real muscles work.

In simple terms, real muscles work by pulling. These muscles are called "skeletal" muscles and they are the ones you're trying to pump when lifting weights, and the ones that are sore when you overwork. Smooth and cardiac muscle are also in our bodies, but are unnecessary when animating computer characters.

Muscles pull by contracting, which shortens and bulges (or squashes) them. When you bend your arm, it is contracting the bicep. In contrast, when you straighten that arm, your triceps pull from the backside of the arm. Though there are many details, the concept of "pulling" will suffice.

Since our muscles are basically attached to our bones, the muscular contraction drives the motion of the bones. When working with "computer" muscles, we must think in reverse. In messiah, the bones drive the motion of the muscles.

Once you understand real muscles you can understand how the muscles on your character should be moving. Character muscles do not work automatically and are depending on you. You don't have to do this for every muscle in your character's body, but you do have to take this job for the muscles that help you pull off your illusion.

Look deeply into messiah, and you'll find a variety of tools for muscle creation. As the creator, use your personal preferences and ideas to select a method that will imbue life into your character. Let's look at those tools.

Standard Morphing

Morphing is probably the best-known technique. Along with standard morphing, messiah offers some alternatives. Morphing moves points, in linear paths, between differently shaped objects. The default object shape is commonly called the base or anchor object and the objects they morph into are called morph targets. Objects are required to have the same point count and order. You can use standard morphing along with the puppet_master effect by morphing your section objects. For standard morphing, messiah offers MorphBlender. Messiah even allows you to import Morph Gizmo files.

Before going further, you should download the supporting files from the Keyframe website, **www.keyframemag.com**.

Load the scene called bicep_morph.fxs. Experiment with this setup by rotating the FOREARM's local pitch and adjusting the bicep_poses slider. This rig requires you to keyframe the bicep_poses slider during an animation, but later you will see how to have messiah do this automatically.

CycleBranchMorphAddFrame

CBMaF is an expression function. It can use stored bone poses to make muscle motion. Its main benefit is that poses are built using bones so it eliminates the need to build morph targets. It can perform everything that morphing can and with the added benefit that points can move in "arcs".

The poses can be controlled by sliders, or automatically with expressions. With either method, you can add additional motion at any time by directly moving or rotating the bones. This function is invaluable for facial animation using bones. I use CBMaF extensively.

Multiple Skeletons and CBMaF

This technique uses two skeletons to deform your character. You can stack more than two skeletons but I have never stacked more than three and normally use just two.

Load bicep_multi.fxs. Look at the hierarchy of bones and you will see two skeletons. One is named "muscle" and the other is "animation". The names describe their functions. Rotate the FOREARM bone and see how clean the deformations look , Figure 1. This is because the multiple skeleton technique gives you vertex-level control with the muscle skeleton while retaining the smooth deformations of the animation skeleton.

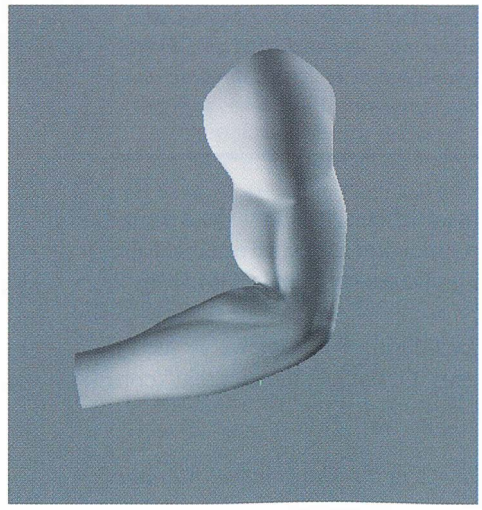


Figure 1: *Clean deformations.*

You can see this technique working with IK in IK_multi_leg.fxs.

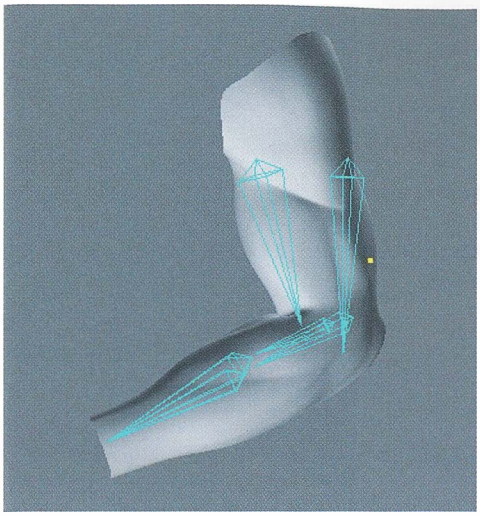


Figure 2: *Muscle Bones*

Muscle Bones

Muscle bones are an ingenious feature of messiah. In essence, they are elastic bones that swell, squash, bulge and stretch just like real muscles do. They are easy to set up, and often all you need. Muscle bones require a bone, a target connection and the use of the Muscle expression.

Load bicep_muscle-bone.fxs. Turn on the muscle bones to see them and then rotate the FOREARM bone. They are light blue and will squash and stretch automatically, Figure 2. Also take a look at a simple example in bicep_simple_musclebone.fxs.

Direct Keyframing

This method does not benefit from expressions or advanced effects. It is simple and direct but can be used when needing a quick result. Put an extra bone to control a muscle and then manually animate it.

Load bicep_direct.fxs. I have keyframed the BICEP bone on the same frame as the FOREARM bone. If the arm bends many times during your animation and you want the bicep to bulge each time, you have to key it every time.

I have described the five techniques I know and use to animate muscles. I'm sure there are others, but I'll emphasize the technique that I feel gives the most power: Multiple Skeletons.

Using Multiple Skeletons: A Primer

For me, the origins of this technique date back to January 2000. Many on the mailing list were discussing the merits of vertex-level control during animations.

Have you ever been working on an animation, and at some point wished that you could just reach into the screen and move just a few points to tweak a deformation? That is the thing we were discussing. We cited a feature called Smartskin in Hash Animation Master. It allows vertex-level control during an animation and is the feature we wanted. While reveling in the control we could get with this, generally under the opinion that it would be the greatest thing since the invention of flight ;) the group received an email from one of the creators of messiah; Lyle Milton.

Lyle made several good points and offered some innovative solutions. I, for one, did not realize that you could use more than one skeleton on an object until then.

He advocated a separate skeleton to "sculpt" muscles. The number of bones needed would vary depending on your needs, but it came down to using just enough bones to achieve your goals. One of the last sentences in that email struck to the heart of the perfectionist residing in me and forms the basis for how I use the Multiple Skeleton technique. This is what he wrote:

"This technique is also good in that you don't have to animate each individual point directly; however, you could if you created a small bone for each point(!)."

... a small bone for each point(!).

That's what he said and then I realized that messiah already provided vertex-level control. I'm not going to tell you to put a separate bone at each vertex of your model because you don't have to. You only use as many bones as you need to accomplish your goal and there will be times when you want a bone at each vertex, which I call vertex_bones, but you only need them in isolated parts of the character.

Some things you can do with multiple skeletons

You've already seen what it can do with bicep_multi.fxs. The muscle skeleton uses vertex_bones for precise control of the bicep. This is how you isolate the technique to specific muscles.

Roughly 80 bones controlling that bicep (Figure 3) and the UPPERARM bone in the animation skeleton controls the main movement of the arm while the vertex_bones in the muscle skeleton move based on the angle of the FOREARM bone.

One skeleton performs its deformations prior to the other. This is what makes the multiple skeleton technique so useful. If I had used one skeleton and added those 80 vertex_bones to the bicep, we would have a bit of a problem. Even though messiah offers greatly improved deformations, putting all those bones parented off the UPPERARM would disrupt the smooth deformation at the elbow joint between the UPPERARM and FOREARM bones.

By putting the vertex_bones in their own skeleton, we allow the UPPERARM and FOREARM to smoothly deform the elbow joint and then we use the vertex_bones to enhance that deformation, not to override it. We can put in the bulge of the bicep, tweak the deformation at the joint, and swell the upper arm to make it look like there is muscle beneath the skin.

The main benefit of using multiple skeletons is keeping all the muscle bones out of the way of the animation skeleton. This is important. The bicep example shows how to achieve very precise control. For most muscles, it could be as simple as adding one or two bones that operate on a muscle area.

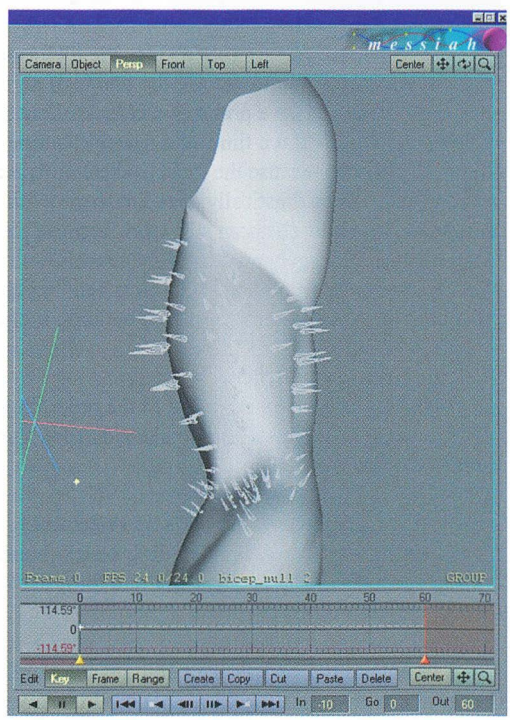


Figure 3: *Eighty bones control bicep*

What about reusability?

Reusability is important because

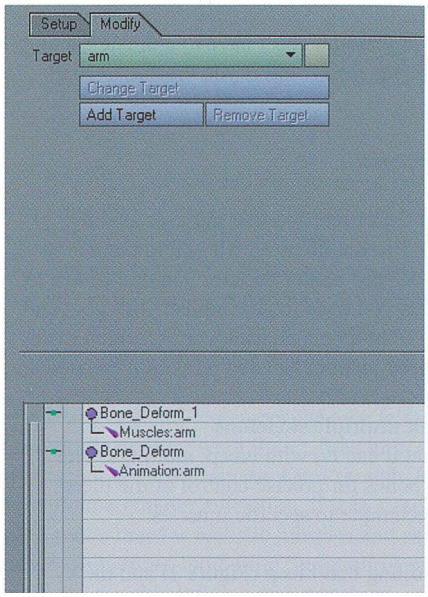


Figure 4: *Ordering effects*

you will save lots of time by animating once. With multiple skeletons, the worst-case is that you would have to rebuild the muscles skeleton for the new character. You will not lose any work from the animation skeleton because it is more generic.

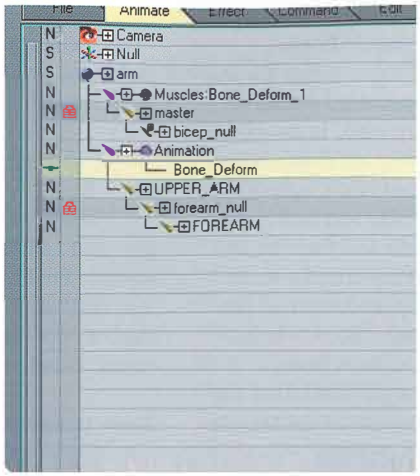


Figure 5: *Use Pre Deformed Points*

I do not think the reusability factor is a big problem. The level of control achieved is a great advantage and I am willing to exchange that for reusability. I want a performance from the character. Other rigging techniques face similar issues. Point-weighting, conditional morphing, and other fine-tuning is usually character specific.

Ordering your Effects

It is important to arrange the skeletons in the correct order. Bone_deform effects can be arranged under the Effects-Modify panel. The muscle skeleton occurs before the animation skeleton (Figure 4).

Messiah has a feature called "Use Pre Deformed Points" (Figure 5). This tells a skeleton to deform the points of an object after they have been deformed by a previous effect. Without this, only one skeleton would work at a time, and it would interfere with morphing and other effects. Because the muscle skeleton is making subtle deformations, it does not drastically alter the shape of the character. The animation skeleton bones can still work because the geometry is still located in the same general area. This also allows muscles shapes to be tweaked while the animation skeleton is bending an arm, and it avoids the laborious trial and error period of building morphs.

Skeleton Activation Order

Skeletons should be activated in the order they are organized in the Effects-Modify panel. Muscle skeleton first, then Animation. Reverse the order when deactivating.

How much time to build the muscle skeleton?

Your time will vary. I find adding vertex_bones and shaping muscle poses faster than going back and forth between Modeler and Layout umpteen times to perfect a morph.

I have minimal experience editing point weights for bone falloff (because I'm always using messiah!) but from my experiences I've also found it to be a trial and error process of many iterations.

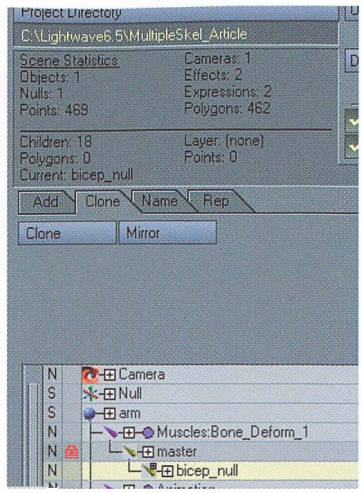


Figure 6: *Clone and Mirror*

It took about 90 minutes to build the muscle skeleton in the bicep example. Once you learn some basics, it's an efficient process.

Building Muscle hierarchies

Bones are built in branching hierarchies to facilitate the use of CBMaF and organization. I usually put the word "null" or "branch" in the first bone of each hierarchy. The branch bones are normally not active and are used as entry points for CBMaF. The bones are organized for ease of use when building muscle shapes. Look at the vertex bones in the bicep example and see how their names reference row and vertex positions.

Orientation

I point the Z-axis of the vertex_bones in the "out" direction. Think of it as being semi-aligned with the surface normal.

The Y-axis is aligned for moving up or down and the X-axis moves side-to-side. This keeps me oriented when building the muscle shapes because of one thing you should understand about using multiple skeletons; when adjusting the muscle bones they are often not in a position that is visually relative to the geometry's current location.

By knowing the orientation of the axes when I first set up the vertex_bones, I can relate this to the proper channel to use. I use the right-mouse-button to move them along local axes. When moving on the Z-axis, I know the vertex is moving "out". Hopefully, that makes some sense but you will need to experiment to learn.

Helpful Hints

I find flat-shaded and wireframe modes to be best for the character. These modes provide a clear picture of where the model vertices are located.

Learn to use the Clone and Mirror feature (Figure 6) when building vertex bones. They will speed up the process and make it easier to place the bones at nearby vertices. Limited Range is also useful for precise control.

How it all works together

A nice thing about messiah is the ease of changing bones after the skeleton is built. You can work on one muscle at a time by testing and refining before moving to the next.

Use the muscle skeleton in specific areas. If you want it for biceps and hamstrings, you only build a skeleton for those areas.

Muscles are animated with sliders or expressions. Expressions allow automatic muscle motion.

In bicep_multi.fxs the default angle of the FOREARM is 0 pitch. At this angle, I want the bulging applied at 0%. I want the expression to apply the muscle shape to 100% as the FOREARM is rotated to 70 pitch. Here's a simple expression that does that:

Bicep_bulging_amount
(RTOD * [FOREARM:pitch]) / 70

Bicep_CBMaF
CycleBranchMorphAddFrame(bicep_null,bicep_null,motion,-10,Bicep_bulging_amount)

I use RTOD so I can work with degrees instead of radians. The first expression calculates the percentage that the FOREARM has been rotated in relation to 70 degrees. If the FOREARM pitch were 57, for example, the value applied to CBMaF would be .81. The muscle shape would be applied at 81% at this point.

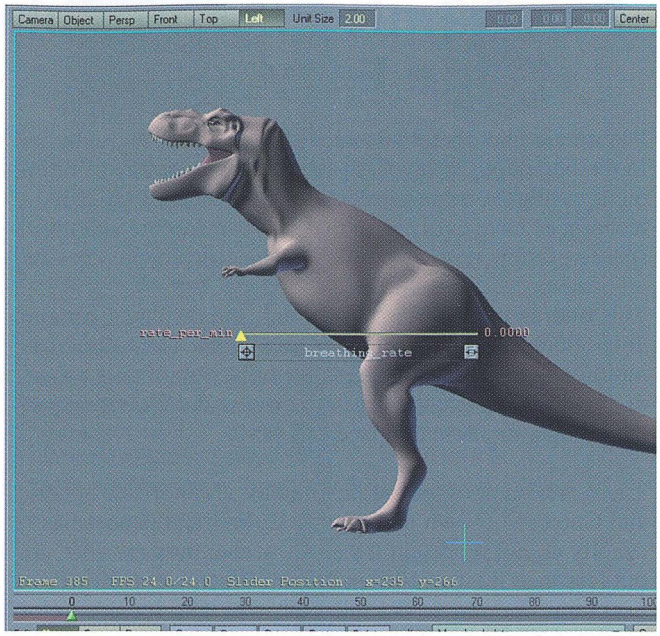


Figure 7: *Autobreathing T. Rex*

In the next issue, we'll look at an example that shows what can be done with the multiple skeleton technique and a little thinking: How to make a T.Rex breathe, realistically and automatically (Figure 7).

Ron Wolfe is a free-lance animator and messiah enthusiast who works out of his home on various projects. He works via the Internet and has recently been animating 3D characters for games with messiah. If you have any questions or comments, please feel free to drop him an email at rwolfejr@woh.rr.com.

Continued from Page 29

Jared's PanoActions
Jared's PanoActions are a series of Photoshop 6.0 actions that automate image conversions. To use these actions, all input (and output files) must be located in the panocon directory on the C: drive (c:\panocon) and match the file names and image resolutions listed. If the resolutions are off, then resize (or stretch) the images to match.

How the actions work: The PanoActions crop, select, and rearrange the panoramic faces. To crop an image numerically, I use Photoshop's Canvas Size command twice. For the first crop, I anchor to the left and get a right edge. For the second, I anchor to the right and get the left edge. The same applies for the top and bottom edges. To assemble the images, I start with a template file that has cubes of alternating solid colors where the faces should be. I then copy the source cubic face to the clipboard. To paste the image to the exact location on the template, I use the magic wand tool and select a solid color cube; I have to be sure I am on the layer with the solid colors. When I select paste, the cubic face gets pasted onto the selection and is perfectly aligned. Since this will create multiple layers, I need to flatten the image when all of the faces are assembled.

Figure 10: *Tallout*

The iPIX Alternative
If you don't want to pay $25/key to save with the iPIX builder software, visit my website at **http://www.jaredjared.com/pano.html** for an alternative procedure that works just as well.

Links
Nikon USA: http://www.nikonusa.com **iPIX**: http://www.ipix.com **Adobe**: http://www.adobe.com
PanoTools: http://www.fh-furtwangen.de/~dersch/ **Bogen**: http://www.bogenphoto.com/
Kaidan: http://www.kaidan.com/ **PTGui**: http://www.ptgui.com/
Apple QuickTime: http://www.apple.com/quicktime/ **MGI Software**: http://www.mgisoft.com/
Apple QuickTime Tools: http://developer.apple.com/quicktime/quicktimeintro/tools/index.html
Philo's Homepage (Panoramic Printing): http://philohome.free.fr/

Jared Bendis is an Instructional Technology Specialist and the Technical Developer for Curricular Virtual Reality at Case Western Reserve University. When he's not inflicting virtual motion sickness on freshmen, he reads email sent to jaredjared@earthlink.net.

Panoramic Photography Made Easy

by Jared Bendis

Panoramic photography is an art as old as photography itself. In the beginning, it involved the careful assembly of multiple pictures aligned edge to edge. Now the technology and techniques have evolved so that you can capture an entire sphere of view with fast, reliable, and reproducible results.

This tutorial outlines how to capture and display a complete 360° (floor to ceiling) spherical panorama. The techniques outlined here are by no means the only way of doing this; they do however represent a time effective and efficient way of capturing a moment of space and time.

There are 3 main types of panoramas: Cylindrical, Spherical and Cubic, as shown in Figure 1. Cylindrical panoramas usually do not represent the entire scene; they are 360° around but less than the full 180° in height. When a cylindrical panorama does cover the full 180° in height, it tends to look distorted. Cylindrical panoramas are easy to create and usually involve just one strip of photographs stitched together.

Spherical panoramas capture the entire 360° around as well as the full height of 180°. Spherical panoramas are difficult to represent as a projection on a plane. One way of projecting a spherical panorama is called an equirectangular projection, Figure 2. This shows the entire 360° width and the 180° height as an image with a 2:1 ratio. However, this causes extreme distortion at the poles. Equirectangular projections are excellent in rendered graphics. When an equirectangular image is spherically mapped to a sphere, it creates a perfect representation of the original view. Additionally, some software packages can read equirectangular projections and display them as spherical panoramas. Lastly, using a variety of tools and filters, equirectangular projections can be transformed, printed, and assembled into spheres or polyhedra that can physically represent the entire scene. For more information on printing techniques visit Philo's Homepage in the links section.

Cubic panoramas also capture the entire 360° around as well as the full height of 180°. Cubic panoramas map the sphere onto 6 sides of a cube. Each side of the cube shows a particular viewing angle and all 6 sides line up at the vertices, Figure 3. The advantages of cubic panoramas are that all 6 sides are individually viewable without distortion around the poles, and they are easy to use in rendered graphics packages as they only require planar projections and planar objects.

The final output of our 360° panoramas will be equirectangular projections, cubic projections, cubic faces, or QuickTime VR panoramas.

The Hardware Setup

Before you can shoot 360° you need to have the right equipment. The easiest way to shoot a 360° panorama is to assemble two images, each with a field of view greater than 180°, that is, images from a spot fisheye lens, as shown in Figure 4. Digital cameras are preferable over film cameras since all of the post-processing is done digitally. Additionally, digital fisheye lenses are less expensive than film lenses and the registration of the spot from image to image is constant, in contrast to the registration of a scanned negative which will vary.

Figure 1: *Panorama types*

Figure 2: *Equirectangular projection*

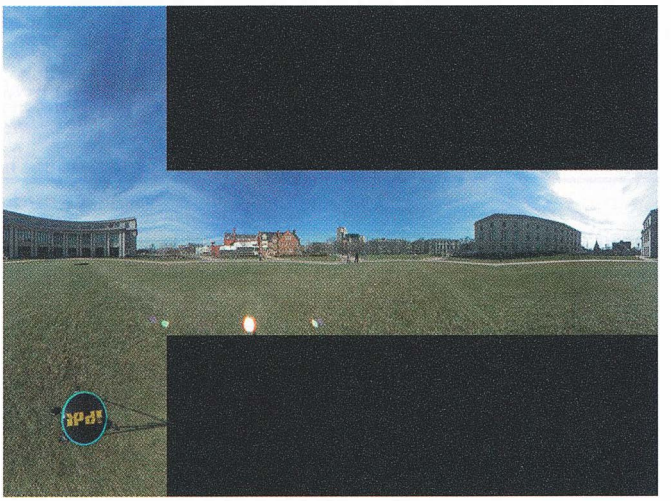


Figure 3: *Cubic projection*

For a Camera/Fisheye combo I use the Nikon 880 with the Fisheye converter lens and step-down adapter ring. This camera allows you to save images as lossless TIFFs (though they are 9MB each). The Fisheye converter lens captures an image with a 183° field of view as a circular spot on the rectangular image.

When you take images for panoramas you need to rotate the camera around the lens' nodal point, *not* around the base of the camera. If you do not rotate around the nodal point you will encounter parallax errors, especially in close shot panoramas. To rotate the camera properly, you need a tripod head that holds the camera by the base but rotates around its nodal point in discreet increments. Panoramic tripod heads range from $150 to $500+, are fully adjustable for a variety of cameras, and all the alignments are done manually. The iPIX Rotator is a custom fit fixed panoramic head sold by iPIX as part of the iPIX Rotator Pack (which also includes the iPIX Builder, now the iPIX 360 Suite software) for $150. The iPIX Rotator is

One of the most versatile plug-ins for LightWave 3D, Shave And A Haircut is now even BETTER! Use it to create, render, and animate hair, grass, wheat, leaves - anything you can imagine, with startling realism. An added bonus is LipService, Joe's incredible facial animation system.....**469.99**

LipService/Shave/Learn LightWave Bundle.....**479.99**

Shadow Designer

A LightWave plug-in that simulates shadows cast from user-defined light sources, independent of LightWave lighting. It darkens pixels lying in shadows, and works with LightWave radiosity and traditional LightWave shadows, as well. It lets you create ANY kind of shadow with ease. If you own LightWave 3D [6] or [6.5] you will want to add this tool to your arsenal.....**99.00**

Available electronically only.

dpsVelocity – 3D

Deliver faster, sharper, easier edits at an affordable price! This real-time editing solution includes the dpsReality card, dpsReality player, V3DX card real-time effect module, audio/ video breakout cable, dps Velocity NLE software, and so much more.....**4699.00**

dpsVelocity – 2D **2899.00**
dpsReality .....**Call**
dpsReality 2D NTSC/PAL **3750.00**
dpsReality Breakout Box.....**899.00**
dpsReality DV Module **1799.00**
dpsVelocity Jog/Shuttle **989.00**
dpsVelocity Tutorial Package ..**40.00**

Visit our site
www.sharbor.com
to build your own ready-to-edit dpsVelocity workstation.

Aura 2

Faster and more powerful!
New features include unlimited video layering, audio import capability, new tools for painting, additional filters, collapsible timelines, transition filters, and so much more.....**649.00**

Aura 2/Illusion Bundle.....**999.00**
Aura 2/FramED Bundle.....**1199.00**
Aura 1.0 - 2 Upgrade.....**295.00**
Inspire 3D/LD_Tools Bundle.....**459.99**
Video Toaster [2].....**4995.00**
Video Toaster NT 1.0 - [2].....**495.00**
LightWave [6.5].....**2250.00**
LightWave [6.5] 3D Studio MAX Competitive Upgrade.....**1189.00**
SX-8 Switcher Expansion Module.....**1995.00**
(Pre-order NOW and get a FREE software upgrade)

We offer a full line of LightWave [6.5] tutorials!

Free T-shirt with select Newtek purchases.

GVX420

Need more power? The GVX420 has dual GLINT Gamma G2 processors, dual GLINT R4 rasterizers, 128MB of on-board memory and flexible analog and DV1 dual display capabilities.....**1699.00**

Oxygen GVX1 Pro AGP
Includes 64MB of on-board memory with virtual texturing **879.00**

Oxygen GVX210 ...

Get 80 photo realistic models with extensive surfacing, multiple layers of maps, in an easy-to-use format. The Botanical is appropriate for beginning LightWave users, and it will meet the needs of high-end advanced users, all of whom will save time without compromising ANY quality.....**125.00**

Looking for a reliable, high-quality video editing or animation workstation? Look no further! We want your buying process to be painless, and that's why we offer our easy on-line workstation configuration system at www.sharbor.com - customize the perfect system for your needs in just a few clicks. Have questions? Just call and talk personally with one of our knowledgeable and committed sales reps. We'll find the right system for your needs at an excellent price.

Our Pentium 4 workstations are fueled by the newest video editing cards from **Pinnacle, Matrox, DPS, NewTek,** and **Canopus.** Safe Harbor workstations feature cutting edge technology like high performance AGP graphics cards, ATA/100 or SCSI-160 video arrays and DVD technology. Every workstation is professionally installed, tested and backed by our superior two-year warranty and our highly rated customer service.

KFramEd

Free 1.5 upgrade now on-line!
A keyframing and compositing solution for Aura. A must have for any Aura user! Features include motion tracking with any number of points, ability to render "on the fly" LightWave objects in layers, ability to create miles-long texts at any size, and much more!..**499.00**

Enlightment (for Aura) **75.00**
PLAB (for Aura) **95.00**

YOUR NUMBER 1

3D Characters Portfolio 1

Over 100 professional 3D Character models ready for you to use in any 3D program capable of importing .LWO or .3DS format. This CD contains everything from wizards to cartoon fruit, and from presidents to dragons. A must have for every animator! **94.99**

3D Characters Portfolio 2

100 more LW [6] character models! This CD contains everything from rock stars to witches, from basketball players to hippos.....**94.99**

3D Characters Portfolio Bundle
Buy both and SAVE! **149.99**

Terms: POs accepted from schools and government agencies.• Personal checks and business checks accepted, but must clear before shipment, usually 10 days • Defective products replaced promptly. RMA number required (call 262-548-8120) for all merchandise returns. Returns not accepted after 15 days. Returned products must be in original packaging, postage prepaid. Opened software not returnable. Shipping charges not refundable. Returns subject to a 15% restocking fee.• Not responsible for typos. Prices subject to change.

Information / Customer Service:
262-548-8120 • Fax 262-548-8130 or
E-mail: customerservice@sharbor.com
Safe Harbor Computers
W226 N900 Eastmound Dr.,
Waukesha, WI 53186

ORDER ON-LINE 24 HRS. A DAY

To Order Call 800-544-6599

MON.– FRI. 9-5:30 CST

SAFE HARBOR COMPUTERS

www.sharbor.com

available in different models for the Nikon Coolpix 700, 800, 880, 950, and 990 cameras. The iPIX Rotator is fixed for a particular model camera, and in addition to rotating around the lens' nodal point also provides extra support for the lens.

The iPIX Rotator has 2 rotational stops exactly 180° apart. I mount my Rotator directly to a basic set of Bogen tripod legs (not to a tripod head or to a cheap tripod) because this removes the need to lock down all the other rotational axes found in a standard tripod head and leaves less visible tripod hardware in the final images. The Rotator, mounted on a basic set of tripod legs, will still show some of the hardware and the iPIX logo that is on the Rotator, but both of these can be edited out later. The full rig cost me about $1250.

Figure 5: *iPIX Rotator, Nikon 880 and Fisheye*

Taking The Images

The Nikon 880 has a built-in setting, Fisheye 1, that automatically locks the camera zoom at maximum and sets the focus at infinity. The hardest part about getting the pictures to look right is matching the lighting in both images. Set the camera in P(rogram)-mode and turn on the Automatic Exposure or AE lock. This sets the camera's exposure and white balance for the first image, then holds them constant for the remaining images. Remember to Reset or turn off the AE lock before shooting the next panorama!

The Automatic Exposure alone is not enough to guarantee a good light and color balance between the two images; you must also use a little common sense. If you do not balance your lighting you will have problems; this is especially true of indoor shots with windows or outdoor shots with the sun shining brightly. The easiest way to balance the lighting is to pick the brightest spot in the scene (the window or the sun) and place the tripod so that the seam of the picture (the far edge of the lens) is on that spot. This will cause the hotspot to be in both pictures and the light balance will be equal for both images. This also means that the hotspot will be on the seam of the final image, and easiest to hide since it will be the most bled out. With multiple hotspots you need to use your best judgment in dividing the lighting of the room. In situations where lighting isn't an issue, such as cloudy days or constant overhead lighting, then you should compose your images so that your seam is at the point of least interest and detail, making it easier to blend.

Beware of your own shadow and reflection. Your shadow will usually show up on the seam, because we placed the hotspot on the seam. The shadow will not be lined up enough to just leave in the final image. To avoid your own shadow (and reflection) take yourself out of the picture entirely by using the auto-timer

Figure 4: *Spot fisheye lens images*

or remote control. The tripod shadow and lens flare from the fisheye will still be in the images and need to be removed manually. It is also important to avoid motion on the seam; you don't want to get half a person who walked away, or swaying trees that won't line up, in the final image.

To completely eliminate the tripod from the final image, you can take a picture of the ground beneath the tripod and later edit it into the final picture. You can also cap the image with a blank space, logo, or a careful reconstruction of what the ground underneath should look like.

At the Shoot:

1) Find spot to shoot from
2) Set seam at hot spot
3) Turn on AE lock
4) Take image 1
5) Rotate 180° (careful to keep tripod steady)
6) Take image 2
7) [Optional] Take image 3 of ground under tripod

The Software Setup

Any version of Photoshop (4.0 and up) will work for this type of image manipulation but my custom Photoshop actions (which are freely downloadable from my website) will only work with Photoshop 6.0. In addition to Photoshop you also need the iPIX Builder software (now the iPIX 360 Suite) to assemble the images. I use the MGI PhotoVista software as a

utility to convert between spherical, cylindrical, and cubic panoramic formats and also as a quick panoramic viewer. If you want to avoid paying the $25/key needed to save the output from the iPIX software, you will also need Helmut Dersch's PanoTools and Joost Nieuwenhuijse's PTGui, both of which are free. All of this software is available for both Windows and Macintosh except for PTGui, which is Windows only.

Converting and Displaying the 360°

To stitch the two images together, the iPIX Builder software needs to know the location and size of the fisheye spots on the source images. The circle that iPIX needs is smaller than the entire spot because the spot captures more than 180°. iPIX has Fisheye Parameter Files on their website for the cameras and Rotators that they sell; but since all cameras are slightly different, the parameters are only a good starting point. You can fine-tune the parameters by hand (a very challenging process) but since we know our images are aligned properly (because of the Rotator) we can let the iPIX software do the work for us. The iPIX software has a tool called Auto Align Digital Camera Images that will attempt to find the size, location, and rotation of the fisheye images. The auto align feature needs to work from images that have some level of details at the seams (unlike regular images that should have as little detail at the seam as possible). It is best to take several different sets of calibration images that have varying levels of details such as ceiling beams, floor tiles, or other information that the software can use for alignment. Once you have aligned the calibration images, save the Fisheye Parameter File. Test the settings by loading in other pairs of fisheye images and the Fisheye Parameter File. If the alignment looks good, click the Save As Default iPIX Parameters icon to lock the settings. If the alignment is off, try different calibration images. If your parameters are good, every time you open two fisheye images in the iPIX builder they should be perfectly aligned.

Figure 6: Saving default parameters

Save the output panorama as an Editable iPIX Image at 1024x2048 resolution using No Compression. I also turn on the Color Balance, Smooth Image, and Blend Seam options; regretfully, you cannot see the results of these options until after you have used a $25 key (if you are unsure, turn off Blend Seam).

The iPIX software installs an import function into Photoshop that imports and edits editable iPIX images. Open Photoshop and select File > Import > iPIX Import. Photoshop loads the panorama in an iPIX viewer that has several options. Click Import Cube and the panorama is converted into a six-cube-wide cubic projection with a resolution of 6138x1023, Figure 8.

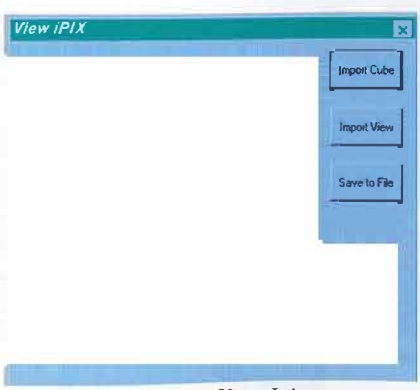


Figure 7: View Ipix

The Photoshop action **iPIX to Faces** (part of Jared's PanoActions) converts the iPIX cubic projection into 6 individual panoramic faces with resolution 1023x1023. In addition to breaking the one image into six, it also re-orders and rotates the images to match the QuickTime standard for cubic image faces.

The image Face6.JPG contains the tripod and tripod head. Edit out or cap the tripod in this image before you continue with the assembly. It is best to work on this image because the projection contains the least amount of distortion and is only a portion of the final image.

The Photoshop action **Faces to PV** (part of Jared's PanoActions) takes the template file tall.psd and the 6 cubic face images and assembles them into a 6 cube tall image with the resolution of 1023x6138, Figure 9. This image is the cubic format used by MGI PhotoVista.

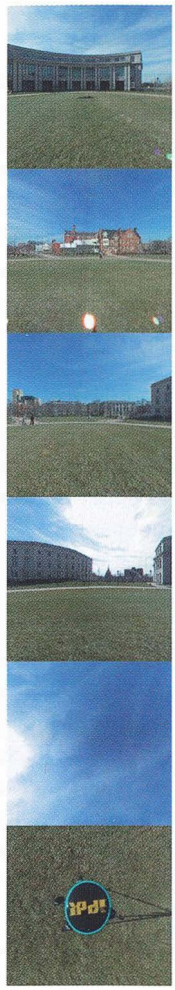


Figure 9

Open MGI PhotoVista, select File > Open Panorama, and load the photovistacube.jpg that you assembled in Photoshop. Select Panorama > Convert to Sphere and it will convert the cubic image into the equirectangular spherical projection. This image can then be saved as an uncompressed JPG. This image should have a resolution of 4080x2040, but will in reality have the resolution of 4080x2032 (somebody stole 8 pixels). At this point I load the image back into Photoshop and resize it from 2032 back to 2040 (without constraining the proportions) to obtain a final equirectangular image of 4080x2040.

An alternate to Faces to PV is the Photoshop action **Faces to PV Tall** (also part of Jared's PanoActions), which takes the template file tall.psd and the 6 cubic face images and assembles them into a six-cube-tall image with a resolution of 1023x6138. This image is the cubic format used by MGI PhotoVista, but it is re-ordered and rotated so that the final equirectangular image is distorted on the right and left instead of the poles. This image needs to be rotated 90° clockwise to be viewed properly (Figure 10, page 25).

The last Photoshop action **PV to Faces** (again, part of Jared's PanoActions) converts the PhotoVista cubic projection into six individual panoramic faces with a resolution of 1023x1023. In addition to breaking the one image into six, it also re-orders and rotates the images to match the QuickTime cubic standard.

Continued on Page 25

Figure 8: iPIX cubic projection

Continued from Page 9

First, you can turn on the Additive option in the Surface Editor panel. This will add the brightness of the transparent object to any surface behind it; so depending on your camera's angle and how the shadow of the droplet is being cast, this may or may not work for you.

Another simple solution is to add a spotlight to your scene and to position it right over the droplet. Adjust the light's cone angle so it only affects the droplet, turn off the shadow options for this light, and then adjust the brightness to a level that gives you the appropriate darkness for the shadow. Again, this solution depends greatly on how the droplet's shadow is falling. It may be necessary to angle the light so that only the droplet's shadow is being lightened and there is no spillover into areas that are not caught in the droplet's shadow. This option will work great for a couple of droplets but it would be a real pain to try to do this for hundreds or thousands of water droplets.

The most realistic and most render-intensive solution is to raytrace shadows and refraction and to also turn on the Caustics feature, which is only available in LightWave 6.5 or later. This will generate shadows that are truly based on the transparency of your water droplet. More importantly, the Caustics feature will bend light based on the Index of Refraction value that you enter for the water droplet's transparent surface. This will result in a cool lens effect where a part of the droplet's shadow has highlights. In this case, it is very important to enter the correct Index of Refraction for water; otherwise, the caustics effect may not look correct. Again, this will give you the most realistic results but you should expect fairly significant render times, depending on the complexity of your scene.

And finally, the most economical solution would be to clone your shadow casting light and turn off all shadow options so you have two lights in the same position shining at the same angle. Adjust the brightness of each light until your scene is lit correctly and the shadows have the darkness level you desire.

For instance, if you set the shadow-casting light's brightness to 30% and set the other light to 60% brightness, you would end up with a nicely lit scene with very soft shadows. This is a very handy way of controlling the intensity of shadows in your scene and, best of all, it will work for any number of objects in your scene.

Q I have a scene where a large, bright, volumetric light is shining directly into the camera, which looks exactly the way I want it to, but when I pass a transparent object in front of the light, the volumetric glow is not visible through the object. Is this a bug or am I doing something wrong?

A I'm not sure if this is a bug or not but the solution is surprisingly simple. All you need to do is set the Refractive Index of your transparent to anything other than 1.0. You can enter a value as small as 1.0001 and the volumetrics will render correctly through the transparent object, as shown in Figure 6, *below*. There is one catch though: in order for this to work, you need to turn on Ray-Traced Refraction, which can add a good bit of render time to your scene, but this will solve the problem. You should also avoid using Shadow-Mapped lights when working with transparent objects, because shadow mapping does not recognize transparent objects and this may have undesirable effects.

See you all again in the next issue!

David Warner is the creator of the TutorVision 3-D series of multimedia tutorial CD-ROMs for LightWave. David has also created a set of searchable archive CDs for the LightWave mailing list that give you quick and easy access to more than 100,000 questions & answers from other LightWave users around the world! For more info and a FREE Sampler CD, please visit ***http://www.tv3d.com*** *or send some e-mail directly to Dave at* ***dave@tv3d.com***

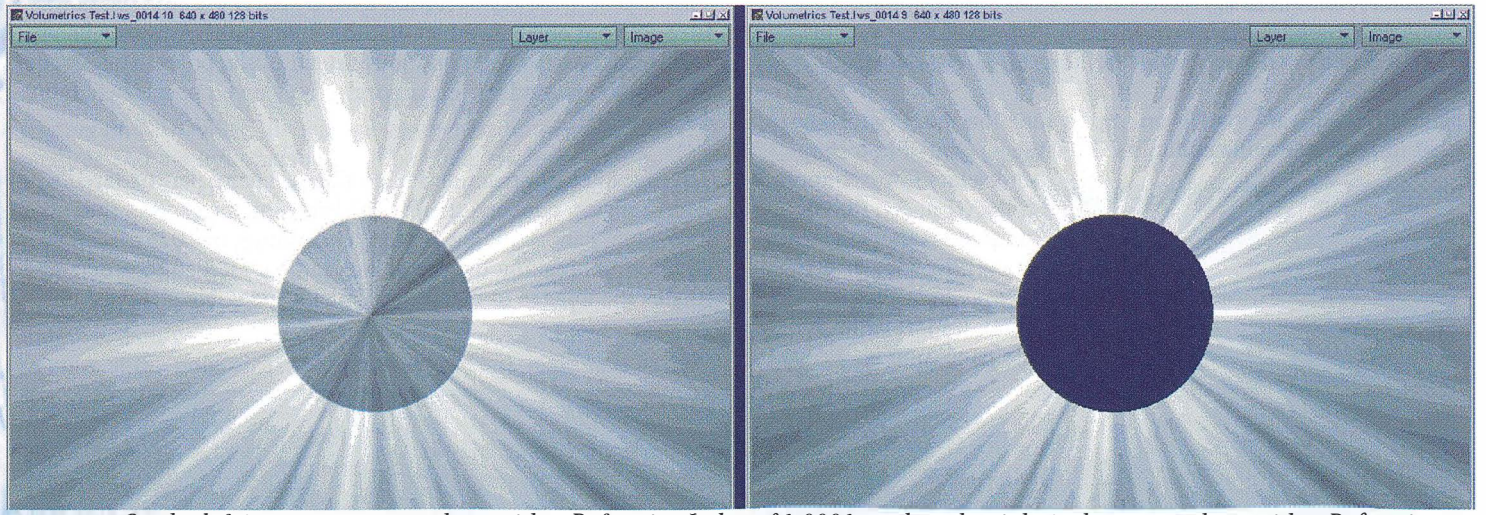


Figure 6: *On the left is a transparent sphere with a Refractive Index of 1.0001, and on the right is the same sphere with a Refractive Index of exactly 1.0. There is a big difference in the rendered results!*

Add a FREE issue...

when you subscribe online at www.keyframemag.com

Require an Invoice?

Print invoices online to qualify for the free issue.

Back issues...

also available online!

KEYFRAME MAGAZINE

DMG Publishing
2756 N Green Valley Pkwy # 261
Henderson, NV 89014
Tel 702-990-8656 Fax 702-616-9647
888-778-WAVE

- ☐ USA 1 year (6 issues).....$54.00
- ☐ Canada/Mexico 1 year $72.00
- ☐ All other countries 1 year.....$84.00

Or subscribe for two years!

- ☐ USA 2 years (12 issues).....$102.00
- ☐ Canada/Mexico 2 years.....$138.00
- ☐ All other countries 2 years.....$162.00

☐ This is a new subscription ☐ This is a renewal

Name _____

Company Name _____

Address _____

City, State and Zip _____

Country _____

Phone _____

Fax _____

Email _____

☐ Charge my card (Visa, MasterCard, American Express, DiscoverCard)

Card # _____ Exp. _____

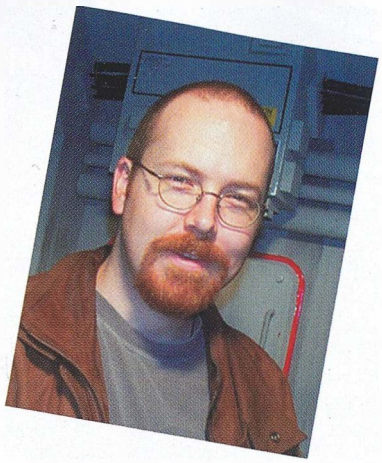
Name on credit card _____

Signature (Required) _____

☐ Check or Money Order payable to DMG Publishing enclosed. US Dollars only.

The Next Five Years

by Steve Pugh



Did you ever wish you had a crystal ball? I guess that's a silly question; after all, who wouldn't want to be able to catch a glimpse of the future? A better question might be, what would you do with it? Sure, any investor worth his weight in failed dot-com stock options would cast a keen eye on next week's S&P Index.

But what about the equally savvy visual effects artist? After the initial rush of learning what the next generation of first-person shooters is going to look like (and, let us be honest, that's always near the forefront of an animator's thoughts), then what? I suppose there are those who would get a kick out of looking far enough into the future to see the Amiga's long-awaited comeback and subsequent dominance of the graphics workstation market during the latter 23rd century. The bundling of Softimage SXI into Microsoft Office 2015 might be good for a giggle or two, with that little paperclip bastard popping up every time you try to change the camera's position.

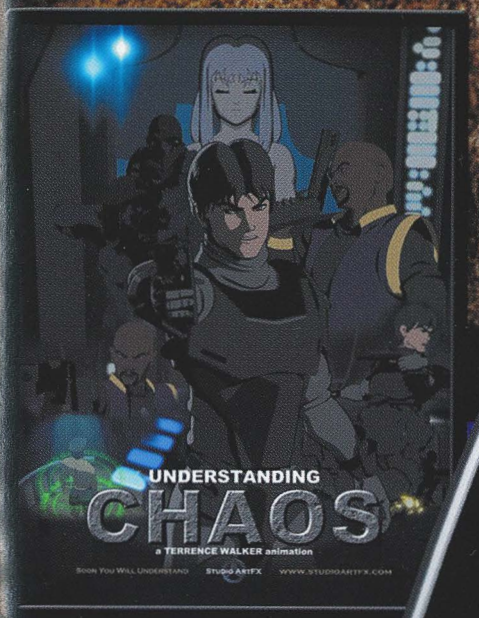
Looking back over the past few years' worth of feature films, though, I'd have to admit that part of me just wants to see what the creative forces come up with next. The fine folks over at Mill Films rebuilt ancient Rome this year, and they took home a statue for it. Used to be, the director would have to settle for a locked-off camera and placing a matte painting behind the actors to convey the massive grandeur of a long-destroyed empire (and by "settle," I certainly don't mean to belittle the significant contributions of the matte painter). However, motion tracking software is now so advanced that the cameraman could shoot every effects sequence duct-taped to the back of a line-dancing dromedary and still marry the performers to a digital set.

A little movie called *The Matrix* came out a while back, and the stylish effects it brought to the screen were so fresh and well-done that they sent a ripple through the whole industry. Unfortunately, they were so fresh that everyone had to put them into their films, sending the bullet-time and frozen-motion effects from 'bleeding edge' to 'bloody well overused' in record time.

I have no doubt that this sort of "new effect out of nowhere" phenomena is only going to grow more frequent as time goes on. The hardware and software are becoming so powerful and so flexible that the best visual effects supervisors (never the sort to sit back and use last year's bag of tricks anyway) are pushing an increasingly broad envelope. Motion tracking and digital set creation/extension are firmly planted in the stock toolkit, and their places on the edge will be filled by a stream of effects meeting that elusive rallying cry of the director, "Something We've Never Seen Before™". Films of sheer hallucinogenic beauty like *The Cell* will become more stunning, more polished, and more commonplace. I hope they'll have better stories, but that's a concern for another day.

Not every effect is going to be a mind-bender, however. This year's *Enemy At the Gates* and *Pearl Harbor* are treating us to skies full of aircraft which haven't flown in decades. They, like *Saving Private Ryan* and *The Patriot* before them, are using visual effects to bring magnitude and gritty impact to stories that center around people rather than the fantastic. While Science Fiction will continue to pop our eyes out and coat them with candy, the use of

Continued on Page 21



The Fastest Video Paint System you've ever seen!

Aura 2

The Desktop Video Paint System

Paint over video and bring life to your animations.

Terrence Walker, creator of **'Understanding Chaos'**, has already discarded all his other 2D and painting tools in favor of Aura's unique flexibility, **"I can't mess with three different packages to get my work done...This one only paints, this other one can composite but you need a third for true 2D animation... Who needs that? With Aura, it's all right in front of me and I can do whatever I need to get the shot done!"**

Whether you are working on print graphics, web animations or video productions, Aura 2 from NewTek is the tool of choice. Aura's next-generation paint engine boasts impressive new features including **four-point pixel tracking, interactive keyframer, stroke recorder, natural media brushes** and **unlimited layers** with **real-time alpha channel**. Enhanced **scripting power** cut hours off production time and adds **flexibility, simplicity** and **speed**. Walker says nothing he's used comes close to matching the speed he's found in Aura. **"Speed is a real issue when you're one person trying to make a real show... Aura is lightning fast!"**

Walker's animation, created in Aura, can be seen at **http://www.studioartfx.com**
Read more about Aura at **http://www.newtek.com**

Creativity without limits is yours with Aura

NewTek
8200 IH-10, Suite 900, San Antonio, Texas 78230 Phone: 220-370-8000 Fax: 210-370-8001
www.newtek.com

Creative imagery by:
TERRENCE WALKER
www.studioartfx.com/uc_view.htm

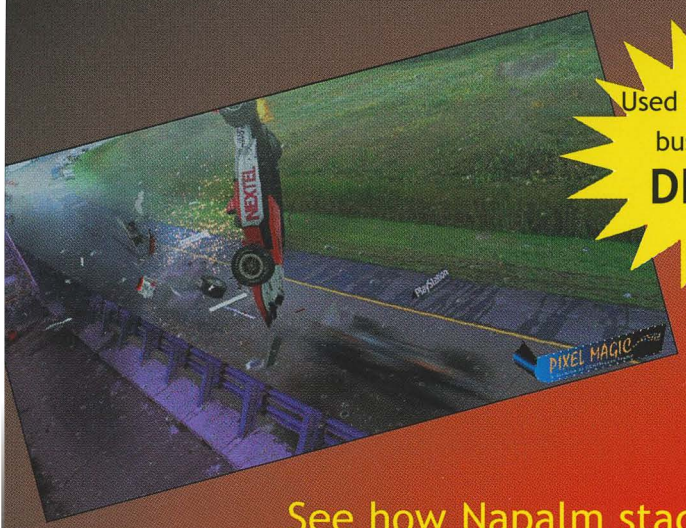
rry!! Price Special!!!
$100 off the normal price
of $495 ($295 upgrade).
Offer expires July 31, 2001

Save More!!!
$200 off Impact 2
(normally $595 retail)
Offer expires July 31, 2001

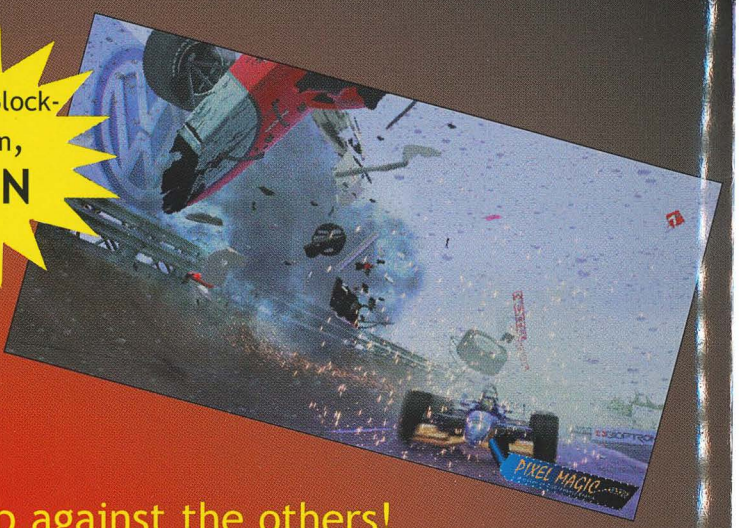
NAPALM

PARTICLE STORM 3

ightWave's most powerful particle system just got better!!!



Used in the Block-buster Film, **DRIVEN**



See how Napalm stacks up against the others!

Feature	LightWave's PFX	Particle Storm 2 Pro	Napalm/Particle Storm 3
Integrated into layout	Yes	No	Yes
React to deformations, including Motion Designer 2K	No	No	Yes
Use the polygons from an object as particles	No	No	Yes
The ability to kill particles in either part or all of scene	No	No	Yes
Interact with Impact v2	No	No	Yes, particles can affect impact objects
Force fields? Drag?	No, No	Yes, Yes	Yes, Yes
Support for weight maps and point groups	No	No	Yes, throughout the program
Envelopable by the graph editor	Yes, but only a few things		Yes, everything can be enveloped and controlled by channel modifiers
Can be controlled by expressions	Yes, but only a few things		Yes, everything can be tied to expressions